

**IMPLEMENTASI LOW POWER PADA SISTEM NOTIFIKASI
UNTUK GAMER YANG BERMAIN *INTENSE* BERBASIS
*EMBEDDED SYSTEM***

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Irfan Pratomo Putra
NIM: 145150300111055



**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

IMPLEMENTASI *LOW POWER* PADA SISTEM NOTIFIKASI UNTUK GAMER YANG
BERMAIN *INTENSE* BERBASIS *EMBEDDED SYSTEM*

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh :

Irfan Pratomo Putra

NIM : 145150300111055

Skripsi ini telah diuji dan dinyatakan lulus pada
29 Juni 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dahnial Syaury, S.T., M.T., M.Sc.
NIK: 2016078704231002

Wijaya Kurniawan, S.T., M.T.
NIP: 19820125 201504 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Sri Astoro Kurniawan, S.T., M.T., Ph.D. A
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, didalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik disuatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 29 Juni 2018

METERAI
TEMPEL

40DB9AFF170397222

6000
ENAM RIBU RUPIAH

Irfan Pratomo Putra

NIM: 145150300111055

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, karena atas berkat dan hidayah-Nya, penulis dapat menyelesaikan dan menyusun laporan skripsi yang berjudul “IMPLEMENTASI *LOW POWER* PADA SISTEM NOTIFIKASI UNTUK *GAMER* YANG BERMAIN *INTENSE* BERBASIS *EMBEDDED SYSTEM*”.

Peneliti menyadari bahwa dalam penyusunan dan penyelesaian laporan skripsi ini tidak akan bisa selesai tanpa mendapatkan bantuan dari pihak-pihak terkait, oleh karena itu peneliti ingin memberikan rasa hormat serta ucapan terima kasih kepada :

1. Bapak Dahnia Syauqy, S.T., M.T., M.Sc. dan Wijaya Kurniawan, S.T., M.T selaku dosen pembimbing yang sudah mau meluangkan waktu, pikiran dan tenaganya dalam membimbing dan mengarahkan peneliti sehingga skripsi ini dapat diselesaikan.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika.
3. Kedua orang tua yang sudah memberikan dukungan kepada peneliti baik dengan do’a maupun dukungan moral.
4. Cipto Bagus Jati Kusumo, Ganda Wibawa Putra serta teman-teman lain sesama peneliti di gedung C yang dengan sabar dan senang hati telah membantu peneliti dalam menyelesaikan skripsi ini baik dengan memberikan saran maupun semangat motivasi
5. Seluruh civitas akademik Fakultas Ilmu Komputer Universitas Brawijaya yang sudah mau membantu peneliti selama menempuh studi di Universitas Brawijaya

Dalam pengerjaan skripsi ini, penulis menyadari bahwa skripsi yang dikerjakan belum sempurna , sehingga untuk segala saran dan kritik akan sangat diterima oleh peneliti. Akhir kata, penulis harap skripsi ini dapat bermanfaat bagi orang yang menggunakannya.

Malang, 29 Juni 2018

Penulis
irfantomi@yahoo.co.id

ABSTRAK

Dewasa ini *game* telah banyak diminati oleh setiap kalangan dari setiap generasi. Khususnya di Indonesia, jumlah pemain *game* sendiri telah mencapai angka 43.7 juta dan 26% di antaranya adalah *gamer* usia 21-35 tahun yang menghabiskan banyak waktu untuk bermain *game* daripada melakukan aktifitas lain. Jika mereka bermain terlalu lama ini akan berbahaya bagi kesehatan mereka terutama kesehatan mata mereka dan bisa menimbulkan kecacatan mata salah satunya adalah cacat rabun jauh. Tidak hanya mata, jika mereka bermain pada ruangan yang kurang pencahayaan, kepala akan terasa pusing. Dengan menggunakan sistem untuk memberikan notifikasi kepada *gamer* yang bermain secara tidak sehat, diharapkan mereka dapat bermain di tempat yang cukup pencahayaan, bermain dalam jarak yang aman dari monitor dan mencegah mereka bermain terlalu lama. Dikarenakan sistem ini akan bekerja dalam jangka waktu yang cukup lama, maka digunakanlah metode *lowpower* untuk mengurangi beban penggunaan sumber daya dari *powerbank*. Kinerja *lowpower* diperlukan agar sistem dapat bertahan lebih lama karena daya yang digunakan relatif rendah ketika sistem sedang tidak digunakan. Fitur dari *lowpower* itu sendiri salah satunya adalah *sleep mode powerdown* yang mematikan beberapa fungsionalitas ATmega328P. Pengujian arus yang dilakukan menghasilkan, bahwa arus yang digunakan pada sistem berjalan normal adalah 169,9 mA sedangkan ketika sistem dalam keadaan *sleep* hanya memiliki arus sebesar 3,8 mA. Selain itu terdapat pengujian waktu yang dibutuhkan sistem untuk *wake* dari keadaan *sleep*, waktu yang dibutuhkan sebesar 22,2 milisekon.

Kata kunci : *Low power, sleep mode power down, notifikasi, game*

ABSTRACT

These days game already in demand by all over society and every generation. Especially in Indonesia, the amount of gamers already reach 43,7 million people with 26% of them is a gamer from 21-35 years old that spend all their time plaining game without do any other activities. If they play for too long this will endanger their helth especially their eye's helth and can cause eye defect of the defect is nearsighted eye's defect. Not only eyes, if they play in a room that don't have enough light, it can cause headache. If they use this system that can give them notification about playing in unhealthy way, i hope they can play in a room that have enough lighting, playing in safe range from monitor, and doesn't play for too long. Because this system will be use for a long time, then lowpower method will be used to reduce the usage of resource from powerbank. The lowpower performance is needed if the system want to hold for awhile longer because the power it use is relatively low if the system is not being use. The feature that lowpower has is sleep mode powerdown that turn off several functionality of ATmega328P. For current testing that already done proof that current that system use when the system function normally is 169,9 mA and for current that system use when the system in sleep mode is 3,8 mA. Beside that, there is a test for time the system take to wake from sleep, the time is 22,2 milisecond.

Keywords : Low power, sleep mode power down, notification, game

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Tinjauan Pustaka.....	6
2.2 Dasar Teori.....	7
2.2.1 Interaksi Manusia Komputer.....	7
2.2.2 Dampak Bermain <i>Game</i>	8
2.2.3 ATmega328P	9
2.2.4 Arduino IDE	11
2.2.5 Kapasitor.....	11
2.2.6 Kristal Osilator	12
2.2.7 Sensor Ultrasonik	12
2.2.8 Sensor Cahaya LDR.....	12
2.2.9 Lampu LED.....	14
2.2.10 Meteran.....	14
2.2.11 Luxmeter	14
BAB 3 METODOLOGI	16

3.1 Studi Literatur	17
3.2 Analisis Kebutuhan	17
3.2.1 Kebutuhan Fungsional.....	17
3.2.2 Kebutuhan Non-Fungsional.....	17
3.3 Desain Sistem	18
3.4 Implementasi	19
3.5 Pengujian dan Analisis	19
3.6 Pengambilan Kesimpulan	20
BAB 4 ANALISIS KEBUTUHAN	21
4.1 Deskripsi Umum Sistem.....	21
4.1.1 Karakteristik Pengguna	21
4.1.2 Ruang Lingkup Sistem	22
4.1.3 Batasan Sistem.....	22
4.2 Kebutuhan Sistem.....	23
4.2.1 Kebutuhan Fungsional.....	23
4.2.2 Kebutuhan Non-Fungsional.....	24
4.2.3 Kebutuhan Peformansi Sistem.....	26
BAB 5 PERANCANGAN DAN IMPLEMENTASI	27
5.1 Perancangan Sistem	27
5.1.1 Perancangan <i>Hardware</i>	28
5.1.2 Perancangan <i>Software</i>	34
5.1.3 Perancangan <i>Prototype</i>	39
5.2 Implementasi Sistem	41
5.2.1 Implementasi <i>Hardware</i>	41
5.2.2 Implementasi <i>Software</i>	45
BAB 6 PENGUJIAN DAN ANALISIS.....	50
6.1 Pengujian Fungsionalitas Sensor Ultrasonik.....	50
6.1.1 Tujuan.....	50
6.1.2 Prosedur Pengujian	50
6.1.3 Hasil dan Analisis.....	51
6.2 Pengujian Fungsionalitas Sensor LDR.....	53
6.2.1 Tujuan.....	54
6.2.2 Prosedur Pengujian	54

6.2.3 Hasil dan Analisis	54
6.3 Pengujian Fungsionalitas <i>Timer</i>	56
6.3.1 Tujuan	57
6.3.2 Prosedur Pengujian	57
6.3.3 Hasil dan Analisis	57
6.4 Pengujian Arus ketika <i>Lowpower Sleep</i> dan <i>Wake</i>	58
6.4.1 Tujuan	58
6.4.2 Prosedur Pengujian	58
6.4.3 Hasil dan Analisis	58
6.5 Pengujian Waktu ketika <i>Wake</i>	60
6.5.1 Tujuan	60
6.5.2 Prosedur Pengujian	60
6.5.3 Hasil dan Analisis	61
BAB 7 PENUTUP	64
7.1 Kesimpulan	64
7.2 Saran	65
DAFTAR PUSTAKA	66



DAFTAR TABEL

Tabel 2.1 Perbedaan sistem dengan penelitian sebelumnya	7
Tabel 2.2 Spesifikasi ATmega328P	Error! Bookmark not defined.
Tabel 4.1 Penjelasan fungsi perangkat keras sistem	24
Tabel 4.2 Fungsi-Fungsi Perangkat Lunak	25
Tabel 5.1 Keterangan Pin Perancangan <i>Hardware</i> ...	Error! Bookmark not defined.
Tabel 5.2 Keterangan Pin Perancangan Catu Daya ...	Error! Bookmark not defined.
Tabel 5.3 Keterangan Pin Perancangan Sensor LDR .	Error! Bookmark not defined.
Tabel 5.4 Keterangan Pin Perancangan Sensor Ultrasonik	32
Tabel 5.5 Keterangan Pin Perancangan LED	33
Tabel 5.6 Implementasi <i>Software Library</i>	46
Tabel 5.7 Implementasi <i>Software</i> Pin-Pin ATmega328P	46
Tabel 5.8 Implementasi <i>Software Low Power</i>	47
Tabel 5.9 Implementasi <i>Software Sub Program</i>	48
Tabel 5.10 Implementasi <i>Software Main Program</i>	Error! Bookmark not defined.
Tabel 6.1 Hasil Pengujian Sensor Ultrasonik	53
Tabel 6.2 Hasil Pengujian Sensor LDR	56
Tabel 6.3 Keberhasilan Fungsionalitas <i>Timer</i> Sistem	57
Tabel 6.4 Perbedaan Arus ketika <i>Sleep</i> dan <i>Wake</i>	60
Tabel 6.5 Hasil Pengujian Waktu <i>Wake</i>	62

DAFTAR GAMBAR

Gambar 2.1 Pin-Pin pada ATmega 328P Sumber : (Atmel, 2008).....	10
Gambar 2.2 Perbedaan <i>Sleep Mode</i> Sumber : (Microchip, 2016)	10
Gambar 2.3 Tampilan Arduino IDE Sumber : (Arduino, 2017).....	10
Gambar 2.5 Kapasitor 22pf Sumber : (Purwanto, 2016)	11
Gambar 2.6 Kristal Osilator Sumber : (Kiswoyo, 2017).....	12
Gambar 2.7 Sensor Ultrasonik Sumber : (depokinstruments, 2006).....	12
Gambar 2.8 Sensor Cahaya Sumber : (zoniaelektro, 2014)	13
Gambar 2.9 Rangkaian Pembagi Tegangan Sumber : (Sparkfun, 2013)	13
Gambar 2.10 LED dan Simbol LED Sumber : (Kho, 2018).....	14
Gambar 2.11 Meteran Sumber : (Kiswari, 2016)	14
Gambar 2.12 Luxmeter Sumber : (Azril, 2012)	15
Gambar 3.1 Diagram Alir Alur Penelitian.....	16
Gambar 3.2 Diagram Blok Sistem	18
Gambar 5.1 Diagram Perancangan Sistem	27
Gambar 5.2 Skematik Perancangan <i>Hardware</i>	28
Gambar 5.3 Skematik Perancangan Catu Daya.....	30
Gambar 5.4 Skematik Perancangan Sensor LDR.....	31
Gambar 5.5 Skematik Perancangan Sensor Ultrasonik.....	32
Gambar 5.6 Skematik Perancangan LED	33
Gambar 5.7 <i>Flowchart</i> Perancangan <i>Software</i> Sistem	34
Gambar 5.8 <i>Flowchart</i> Perancangan Sensor LDR.....	35
Gambar 5.9 <i>Flowchart</i> Perancangan <i>Software</i> Sensor Ultrasonik.....	36
Gambar 5.10 Perancangan <i>Software Timer</i>	37
Gambar 5.11 Perancangan <i>Software Low Power</i>	38
Gambar 5.12 Skematik Perancangan PCB.....	40
Gambar 5.13 Perancangan Aklirok Tampak Depan (a) Tampak Atas (b)	41
Gambar 5.14 Implementasi <i>Hardware</i> PCB	42
Gambar 5.15 Implementasi <i>Hardware Prototype</i> Tampak Atas.....	42

Gambar 5.16 Implementasi Hardware Prototype Tampak Depan	43
Gambar 5.17 Implementasi Catu Daya	43
Gambar 5.18 Implementasi <i>Hardware</i> Sensor LDR	44
Gambar 5.19 Implementasi <i>Hardware</i> Sensor Ultrasonik	45
Gambar 5.20 Implementasi <i>Hardware</i> LED	45
Gambar 6.1 Pengujian Sensor Ultrasonik pada <i>Serial Monitor</i>	52
Gambar 6.2 Pengujian Sensor Ultrasonik pada Meteran	52
Gambar 6.3 Pengujian Sensor LDR pada <i>Serial Monitor</i>	55
Gambar 6.4 Pengujian Sensor LDR pada Luxmeter	55
Gambar 6.5 Hasil Pembacaan Arus oleh Multimeter ketika <i>Sleep</i>	59
Gambar 6.6 Hasil Pembacaan Arus oleh Multimeter ketika <i>Wake</i>	59
Gambar 6.7 Hasil Waktu <i>Wake</i> pada <i>Serial Monitor</i>	61
Gambar 6.8 Penggalan <i>Source Code</i> Pertama	62
Gambar 6.9 Penggalan <i>Source Code</i> Kedua	62



BAB 1 PENDAHULUAN

Pada bab 1 pendahuluan dipaparkan mengenai awal mula penelitian ini dibuat. Dimulai dari latar belakang dari penelitian, lalu dari latar belakang akan menghasilkan rumusan masalah, berikutnya tujuan penelitian yang merupakan jawaban dari rumusan masalah, selanjutnya manfaat penelitian, batasan masalah, dan terakhir sistematika pembahasan.

1.1 Latar Belakang

Dewasa ini, *game* sudah banyak diminati oleh setiap kalangan dan setiap generasi. Baik anak-anak, remaja, hingga orang dewasa sangat menggemari *game*. Berdasarkan survey dari newzoo pada tahun 2017, total *gamer* yang ada di Indonesia sudah mencapai angka 43.7 Juta orang dimana 26%-nya adalah *gamer* usia 21-35 tahun (Newzoo, 2017). Bahkan anak-anak usia 2-12 tahun menghabiskan banyak waktu untuk bermain *game* dari pada melakukan aktifitas lain (NPD, 2015). Jelas ini akan berbahaya untuk kesehatan mereka, terutama mata mereka, terlalu lama bermain *game* akan membuat mata terasa perih dan dapat menimbulkan kecacatan mata, misalnya mata menjadi cacat rabun jauh. Tak hanya mata, bagian-bagian lain seperti tangan dan pinggang juga pasti mengalami kelelahan misalnya pegal-pegal, kaku, hingga keram. Terkadang terlalu banyak bermain *game* juga bisa membuat kepala terasa pusing terlebih jika bermain pada ruangan yang kekurangan cahaya (Wind, 2016).

Pemerintah sudah mencoba untuk mencegah korban meningkat, salah satu cara yang diterapkan di Indonesia adalah mengancam akan memblokir beberapa *game* yang mengandung unsur kekerasan (Kemendikbud, 2016). Namun usaha dari pemerintah ini terlihat tidak begitu berpengaruh, karena para *gamer* masih saja memainkan *game* tersebut walaupun mengandung unsur kekerasan. Peran orang tua menjadi salah satu peran yang sangat penting agar banyak anak-anak dan remaja tidak ketagihan *game*, namun terkadang orang tua sibuk dengan pekerjaan mereka sehingga anak-anak mereka tidak diperhatikan.

Penggunaan sumber daya juga menjadi salah satu permasalahan yang terjadi seiring dengan berkembangnya teknologi. Terutama pada perangkat sistem cerdas dimana memiliki prinsip kerja bekerja terus menerus dengan sumber daya yang terbatas, lalu pada penggunaannya membutuhkan pemantauan terus menerus. Ini akan menjadi utama faktor dalam pemborosan sumber daya (Firmansyah, 2018).

Berdasarkan permasalahan yang telah disebutkan di atas, penulis bermaksud melakukan penelitian yang berjudul "IMPLEMENTASI LOW POWER PADA SISTEM NOTIFIKASI KEPADA GAMER YANG BERMAIN INTENSE BERBASIS EMBEDDED SYSTEM" dimana tujuan dari sistem ini dapat memberikan notifikasi kepada *gamer* yang bermain lebih dari batas normal dan tidak wajar namun konsumsi daya sistem yang digunakan kecil (*Low*). Peringatan ini bisa dengan peringatan

pencahayaan dengan menggunakan sensor cahaya LDR (*Light Dependent Resistor*), dimana *gamer* harus bermain di ruangan yang pencahayaannya cukup terang sehingga tidak menimbulkan pusingnya kepala. Selain itu, ada peringatan yang akan memperingati *gamer* untuk bermain pada jarak yang normal dengan menggunakan sensor ultrasonik, tidak terlalu dekat dan juga tidak terlalu jauh, dengan harapan dapat mencegah mata mengalami cacat rabun jauh maupun kecacatan lainnya. Lalu, peringatan yang akan memperingati tentang berapa lama *gamer* telah bermain *game* agar *gamer* tidak bermain dalam jangka waktu yang terlalu lama dengan menggunakan *timer*. Ketiga parameter tersebut akan memberikan notifikasi hanya dengan menggunakan fitur *If-Else*. *Gamer* dapat dikatakan bermain secara *intense* jika satu, dua, atau tiga dari LED yang bersangkutan berkedip.

Lalu sesuai yang telah disebutkan bahwa sistem akan menggunakan sedikit daya. Penulis menggunakan metode *low power* agar sistem dapat bekerja dalam jangka waktu yang cukup lama (menghemat penggunaan sumber daya), bekerja efisien (konsumsi energi yang sedikit), dan bekerja efektif (arus dan tegangan sistem akan rendah ketika sistem sedang tidak digunakan). Penerapan *Low Power* diimplementasikan dengan adanya fitur *Sleep* dan *Wake*, dimana pada fitur *Sleep* akan aktif ketika pengguna sedang menjauh dari sensor ultrasonik (misalnya pergi makan siang) dimana sistem akan menghentikan pengambilan data dari sensor disebabkan dimatikannya CPU sistem. Walaupun sistem dalam keadaan *Sleep* sistem juga dapat diaktifkan kembali dengan menggunakan *external interrupt* yang ada difitur *Wake*.

Notifikasi yang ada pada sistem cukup menggunakan LED (*Light Emitting Diode*) dimana akan berkedip jika pengguna bermain di tempat gelap atau bermain pada jarak yang dekat, serta bermain melebihi batas waktu yang telah ditentukan. Diharapkan sistem yang diteliti dapat bekerja dengan baik.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka dapat dirumuskan masalah seperti berikut :

1. Bagaimana ketiga Parameter dapat menentukan notifikasi kepada pengguna?
2. Bagaimana akurasi sensor ultrasonik dan sensor LDR dalam mengambil data dengan alat pengukur jarak dan Luxmeter?
3. Bagaimana mengimplementasikan *low power* pada sistem notifikasi kepada pengguna?

1.3 Tujuan Penelitian

Berdasarkan Rumusan Masalah yang telah diuraikan di atas, maka didapatkan tujuan dari penelitian sebagai berikut :

1. Menguji akurasi sensor ultrasonik dan sensor LDR dalam mengambil data dengan alat pengukur jarak dan Luxmeter.
2. Ketiga parameter dapat menentukan notifikasi kepada pengguna.
3. Mengimplementasikan *low power* pada sistem notifikasi kepada pengguna berdasarkan parameter cahaya, jarak, dan waktu.

1.4 Manfaat Penelitian

Penelitian ini juga memiliki manfaat dimana secara praktis, manfaat penelitian ini dapat dilihat pada poin-poin berikut :

1. *Gamer*, dapat mengetahui sudah berapa lama mereka bermain game, mengetahui apakah ruangan sudah cukup pencahayaannya, dan juga mengingatkan mereka untuk menjaga jarak antara mata dengan layar monitor/laptop .
2. Orang Tua, dapat membuat mereka lebih tenang jika anak mereka adalah seorang *gamer* .
3. Masyarakat, dapat meyakinkan mereka bahwa bermain *game* bisa dilakukan secara sehat.
4. Peneliti, karena peneliti adalah seorang *gamer* maka penelitian ini akan sangat berdampak baik kepada peneliti itu sendiri.

Manfaat penelitian ini secara teoritis adalah :

Secara teoritis, hasil penelitian ini dapat bermanfaat kepada peneliti lain yang ingin meneliti lebih lanjut terhadap kesehatan *gamer* serta bahaya lainnya.

1.5 Batasan Masalah

Batasan masalah meliputi batasan-batasan dari sistem yang diteliti. Batasan-batasan dari permasalahan di atas meliputi :

1. Sistem diujikan kepada remaja sehat dengan usia 22 Tahun.
2. Sistem diujikan hanya menggunakan 3 parameter (Cahaya, Jarak, dan Waktu).
3. Sistem memberikan notifikasi dalam bentuk nyala lampu LED.
4. Sistem berada di dalam ruangan tertutup
5. Sistem diujikan kepada pengguna yang duduk di kursi dan monitor di atas meja.
6. Sistem diujikan menggunakan metode *Low Power*.
7. Sistem berbentuk *portable* dan bukan *wearable*.
8. Monitor yang digunakan adalah monitor LCD (*Liquid Crystal Display*) 24 inch.

1.6 Sistematika Pembahasan

Sistematika pembahasan dari klasifikasi peringatan kepada *gamer* yang bermain *intense* menggunakan *naive bayes* berbasis *embedded system* adalah sebagai berikut :

BAB 1 PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, dan batasan masalah dan sistematika pembahasan.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dijelasakan mengenai studi literatur, dasar teori dari penelitian peringatan kepada *gamer* yang bermain *intense* dengan mengimplementasikan *Low Power* berbasis *embedded system*, serta jenis-jenis sensor yang digunakan.

BAB 3 METODOLOGI

Pada bab ini akan dijelaskan mengenai metode dan langkah-langkah yang dilakukan dalam pengerjaan penelitian ini. Dimulai dari studi literatur, analisis kebutuhan, desain sistem, implementasi, pengujian dan analisis, dan pengambilan kesimpulan dari penelitian peringatan kepada *gamer* yang bermain *intense* dengan mengimplementasikan *Low Power* berbasis *embedded system*.

BAB 4 ANALISIS KEBUTUHAN

Pada bab ini akan dijelaskan mengenai kebutuhan-kebutuhan sistem agar sistem dapat berjalan dengan baik dan sesuai harapan. Bab ini berisi deskripsi umum sistem, rekaya kebutuhan, kebutuhan fungsional, kebutuhan non-fungsional dimana berisi kebutuhan perangkat keras dan kebutuhan perangkat lunak, lalu kebutuhan performansi sistem.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai perancangan dari sistem dimulai dari perancangan sistem secara *hardware*, *software*, hingga *prototype* dari sistem itu sendiri. Setelah sistem dirancang maka akan diimplementasikan ke dalam bentuk aslinya dimana pada *hardware* sistem yang diimplementasi ke dalam PCB dan juga sistem sudah menjadi *portable*. Untuk implementasi *software* adalah sistem sudah bisa menjalankan fungsionalnya dengan kode program yang telah diunggah.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dijelaskan mengenai bagaimana sistem dapat diuji tingkat keberhasilan dan fungsionalitasnya. Tingkat

fungsi dan keberhasilan sistem diuji pada sensor ultrasonik, sensor LDR, dan *timer*. Selain itu, sistem juga diuji mengenai pemakaian arus ketika sistem dalam keadaan *sleep* dan dalam keadaan *wake*. Sistem juga diuji mengenai waktu bangun sistem dari keadaan *sleep* dalam satuan milisekon.

BAB 7 PENUTUP

Pada bab ini akan dipaparkan mengenai kesimpulan dan saran dari sistem yang diteliti. Kesimpulan akan diambil inti dari keseluruhan sistem, lalu pada saran akan diberikan peneliti kepada peneliti lain ketika sistem ini ingin dikembangkan lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab 2 landasan kepastakaan, terdapat sub bab tinjauan pustaka yang berisi informasi dari penelitian-penelitian lain yang berhubungan dengan penelitian ini. Lalu sub bab berikutnya adalah sub bab dasar teori yang berisi teori-teori dasar untuk penelitian ini.

2.1 Tinjauan Pustaka

Dalam penulisan skripsi, peneliti juga menggali informasi mengenai penelitian-penelitian sebelumnya yang relevan sebagai bahan referensi. Jurnal Eka Rusnani Fauziah, Mahasiswi Universitas Mulawarman, Samarinda yang berjudul "PENGARUH GAME ONLINE TERHADAP PERUBAHAN PERILAKU ANAK SMP NEGERI 1 SAMBOJA" Jurnal ini membahas tentang game online Ragnarok terhadap perilaku remaja dimana pengaruh tersebut memiliki pengaruh positif dan negatif. Perilaku positif adalah mereka dapat berdiskusi dengan orang lain sehingga menambah teman mereka, negatifnya mereka akan menjadi malas belajar (Fauziah, 2013).

Skripsi Eko Hilmi Firmansyah, Mahasiswa Universitas Brawijaya, Malang yang berjudul "Implementasi Low Power Mode Pada Perangkat Sistem Pendeteksi Dini Kebocoran Gas Menggunakan ATmega328p" penelitian terkait dengan sistem untuk mendeteksi kebocoran gas. Agar perangkat sistem berjalan secara efisien, maka perangkat dalam penerapannya menggunakan kinerja low power untuk menghemat konsumsi daya. Penghematan dilakukan agar perangkat dapat bekerja dalam jangka waktu relatif panjang menggunakan sumber energi baterai. Memanfaatkan kinerja low power pada mikrokontroler ATmega328p diharapkan memberikan implementasi kebutuhan perangkat pada konsumsi daya yang lebih hemat (Firmansyah, 2018).

Jurnal Ganjar Turesna, Zulkarnain, Hermawan dari Teknik Elektro Universitas Langlang Buana yang berjudul "Pengendali Intensitas Lampu Ruangan Berbasis Arduino UNO Menggunakan Metode Fuzzy Logic" yang membahas saat pengguna lampu dalam ruangan menjalankan sistem atau menyalakan lampu maka sensor cahaya LDR menerima cahaya dari luar yang menyebabkan perubahan level tegangan input ke mikrokontroler yang selanjutnya diproses dengan output mikrokontroler berupa tegangan pulsa (PWM) untuk menyalakan lampu LED (Turesna, Zulkarnain, & Hermawan, 2015) .

Berdasarkan penelitian di atas, dapat diketahui bahwa penelitian mengenai metode *Low Power*, mengenai *game*, dan juga mengenai intensitas lampu sudah pernah dilakukan. Berikut merupakan perbedaan antara penelitian sebelumnya dengan penelitian sekarang, dapat dilihat pada tabel 2.1.

Tabel 2.1 Perbedaan sistem dengan penelitian sebelumnya

No	Penulis, Judul sebelumnya	Penelitian Sebelumnya	Perbedaan
1	Eka Rusnani Fauziah, Pengaruh Game Online Terhadap Perubahan Perilaku Anak Smp Negeri 1 Samboja	Pengaruh <i>game online</i> kepada perilaku remaja, apakah berpengaruh positif atau negatif	Sistem menggunakan <i>Embedded System</i>
2	Eko Hilmi Firmansyah, Implementasi Low Power Mode Pada Perangkat Sistem Pendeteksi Dini Kebocoran Gas Menggunakan ATmega328p	Perangkat cerdas untuk mendeteksi kebocoran dengan menggunakan metode <i>Low Power Sleep Mode Power Down</i>	Perangkat cerdas untuk mencegah <i>gamer</i> bermain berlebihan
3	Ganjar Turesna, Zulkarnain dan Hermawan, Pengendali Intensitas Lampu Ruangan Berbasis Arduino UNO Menggunakan Metode Fuzzy Logic	Menggunakan sensor LDR untuk mengendalikan intensitas lampu ketika pengguna sedang menggunakan lampu atau lampu sedang menyala	Menggunakan sensor LDR untuk mendeteksi apakah pengguna sedang menggunakan komputer dengan kadar intensitas cahaya yang telah ditentukan

2.2 Dasar Teori

Dalam Sub-bab ini berisi penjelasan mengenai teori-teori yang digunakan dalam penelitian ini diantaranya adalah, Algoritma *Naive Bayes*, Sensor Ultrasonik HC-05, Sensor Cahaya LDR, Mikrokontroler Arduino, LCD 16x2, dan Interaksi Manusia Komputer.

2.2.1 Interaksi Manusia Komputer

Interaksi Manusia Komputer atau IMK adalah suatu ilmu untuk mempelajari mengenai perancangan, implementasi, dan evaluasi suatu sistem agar sistem tersebut dapat dengan mudah digunakan oleh manusia. IMK juga bisa dikatakan sebagai acuan mengenai bagaimana interaksi antara manusia dan komputer dapat terlibat. Fokus dari IMK itu sendiri adalah bagaimana satu manusia atau lebih dapat berinteraksi dengan baik terhadap 1 komputer atau lebih, contohnya adalah bagaimana manusia menggunakan laptop mereka, bagaimana suatu grup manusia menggunakan berbagai macam komputer untuk mengerjakan proyek

mereka, bagaimana satu manusia dapat menangani berbagai macam komputer dan lain sebagainya.

Istilah “Interaksi Manusia dan Komputer” atau “Interaksi Manusia dan Mesin” melingkupi dua sisi, yaitu mesin dan manusia. Pertama dengan melihat istilah “mesin”, istilah ini lebih populer dengan sebutan komputer. Berbagai jenis komputer yang kita kenal antara lain adalah *mainframe*, *workstation*, dan komputer pribadi (PC/Laptop). Selain berbagai jenis di atas komputer juga muncul dalam bentuk mesin komputasi terpadu, misalnya mesin cuci, *microwave*, *printer*, dan sebagainya (Santoso, 2009). Dengan demikian suatu komputer atau mesin harus dirancang antar-mukanya sebagaimana rupa agar mesin tersebut dapat dipahami dengan baik oleh manusia yang menggunakan mesin tersebut.

Berikutnya adalah manusia, karena komputer telah memberikan suatu antarmuka maka diharapkan manusia dapat mengenali, dan mempelajari dalam penggunaan komputer/mesin tersebut. Meskipun begitu tidak menutup kemungkinan pengguna bisa melakukan suatu kesalahan, baik kesalahan kecil maupun kesalahan besar, baik kesalahan yang disengaja atau tidak disengaja. Hal ini disebut sebagai *human error* (Santoso, 2009).

Kedua aspek di atas termasuk aspek penting dalam mencapai Interaksi yang baik antara Manusia dan Komputer.

2.2.2 Dampak Bermain *Game*

Arti *game* dalam Bahasa Indonesia adalah permainan, dimana *game* memiliki fungsi untuk menghibur penggunanya yang disebut dengan *gamer* ketika sedang bosan atau saat waktu luang. Hiburan yang diberikan oleh media *game* berbagai macam, ada yang *game* yang bisa dilakukan secara *online* dan ada *game* yang dapat dilakukan secara *offline*. Meskipun fungsi dari *game* itu sendiri adalah untuk menghibur, *game* memiliki dampak kepada *gamer* baik itu dampak baik maupun dampak buruk. Untuk dampak baiknya jelas *game* dapat membuat penggunanya merasa terhibur ketika sedang bosan. Selain itu *gamer* juga dapat mengasah kecerdasan mereka dengan bermain *game* berbasis *strategy* seperti misalnya catur. *Game* juga dapat membuat para pengguna sesama *gamer* saling berkomunikasi dan saling bertukar informasi tentang *game* yang mereka mainkan sehingga ini akan menambah teman mereka untuk bersosialisasi (Anggraini, 2016).

Selain dampak baik, *game* juga akan memberikan dampak buruk kepada *gamer*. Salah satunya adalah kecanduan *game*. *Gamer* yang bermain *game* secara *intense* dimana *intense* sendiri dalam Bahasa Indonesia adalah giat, merupakan salah satu pemain *game* atau *gamer* yang sudah kecanduan dalam bermain *game*. Sering sekali *gamer-gamer* kecanduan *game*, terutama *game online*. Efek dari kecanduan sangat berbahaya bagi kesehatan mereka. Seperti pada mata mereka, *gamer* yang kecanduan bermain *game* cenderung untuk memaksa mata mereka untuk melihat layar monitor walaupun mereka sudah

lelah/mengantuk, terlebih lagi jika mereka bermain pada ruangan yang kurang pencahayaan atau mereka bermain pada jarak antara mata dengan monitor yang terlalu dekat. Jika mereka terus memaksakan, maka akan membuat mata mereka cacat karena terlalu sering menerima radiasi layar monitor. Salah satu kecacatannya adalah rabun jauh. Selain berdampak pada mata mereka, *game* juga dapat berdampak pada psikologis mereka. Orang yang kecanduan bermain *game* cenderung malas belajar dan sulit berkonsentrasi, mereka juga tidak peduli, cuek, acuh tak acuh terhadap lingkungan mereka (Efendi, 2014).

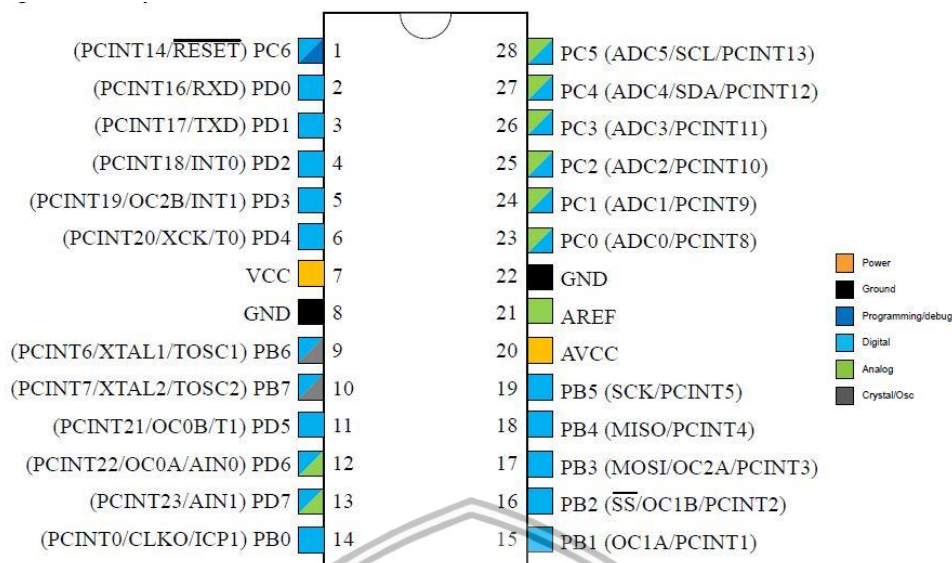
2.2.3 ATmega328P

ATmega328P adalah suatu *chip* mikrokontroler 8-bit yang menjadi unit pemrosesan, atau bisa dikatakan, ATmega328P adalah otak dari sistem ini. ATmega328P buatan Atmel, dimana *chip* ini memiliki 32 KB memori ISP flash dengan kemampuan baca-tulis (read write), 1 KB EEPROM, dan 2 KB SRAM (Atmel, 2008). Berikut merupakan ringkasan spesifikasi ATmega328P yang ditunjukkan pada Tabel 2.1.

Tabel 2.2 Spesifikasi ATmega328P

Ringkasan Spesifikasi ATmega328P	
Program Memory Type	Flash
Program Memory dalam KB	32
Kecepatan CPU dalam MIPS	20
RAM Bytes	2048
EEPROM dalam KB	1
SRAM dalam KB	2
Register	32
Jumlah Programmable I/O	32
Jumlah Timer/Counter	3
Perperal PWM	6
Range Temperatur dalam Celcius	-40 sampai 85
Voltase	1.8V – 5.5V

Untuk lokasi dan keterangan mengenai pin-pin yang ada pada ATmega328P, ditunjukkan pada Gambar 2.1.



Gambar 2.1 Pin-Pin pada ATmega 328P

Sumber : (Microchip, 2016)

Selain yang telah disebutkan di atas, terdapat fitur-fitur lain yang ditawarkan oleh ATmega328P, salah satunya adalah fitur mode *Sleep* dimana terdapat berbagai jenis mode *Sleep* yang ditawarkan, yaitu *idle*, *ADC Noise Reduction*, *Power-Down*, *Power-Save*, *Standby*, dan *Extended Standby* yang mengacu pada Gambar 2.2.

Sleep Mode	Active Clock Domains					Oscillators		Wake-Up Sources						
	clk _{CPU}	clk _{FLASH}	clk _{I/O}	clk _{ADC}	clk _{ASY}	Main Clock Source Enabled	Timer Oscillator Enabled	INT and PCINT	TWI Address Match	Timer2	SPM/EEPROM Ready	ADC	WDT	Other I/O
Idle			Yes	Yes	Yes	Yes	Yes ⁽²⁾	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ADC Noise Reduction				Yes	Yes	Yes	Yes ⁽²⁾	Yes ⁽³⁾	Yes	Yes ⁽²⁾	Yes	Yes	Yes	
Power-Down								Yes ⁽³⁾	Yes				Yes	Yes
Power-Save					Yes		Yes ⁽²⁾	Yes ⁽³⁾	Yes	Yes			Yes	Yes
Standby ⁽¹⁾						Yes		Yes ⁽³⁾	Yes				Yes	Yes
Extended Standby					Yes ⁽²⁾	Yes	Yes ⁽²⁾	Yes ⁽³⁾	Yes	Yes			Yes	Yes

Gambar 2.2 Perbedaan Sleep Mode

Sumber : (Microchip, 2016)

Setiap mode *Sleep* memiliki berbagai macam fungsi yang aktif dan di non-aktifkan. Misalnya pada mode *sleep power down* yang mematikan fungsi *active clock domains*, *timer oscillator*, *timer*, *SPM/EEPROM*, *ADC*, dan *BDO*. Contoh lain adalah *sleep idle* hanya fungsi *Clock CPU*, *Clock Flash*, dan *BOD Disable* saja yang di non-aktifkan, sisanya aktif. Peneliti menggunakan *sleep power down* karena mode *sleep* inilah yang paling banyak mematikan fungsional dari ATmega328P sehingga akan menghemat penggunaan daya ketika sistem masuk ke mode *sleep*.

2.2.4 Arduino IDE

Untuk mengoding ATmega328P, dapat menggunakan *software* Arduino IDE, dimana *software* ini menggunakan bahasa C++ yang bisa *download* di situs arduino. Berikut merupakan tampilan Arduino IDE setelah selesai di *install*. Tampilan Arduino IDE ditunjukkan pada Gambar 2.3.



Gambar 2.3 Tampilan Arduino IDE
Sumber : (Arduino, 2017)

Terdapat 2 fungsi utama untuk memprogram ATmega328P agar program dapat dijalankan, yaitu void setup() dan void loop(). Void setup () adalah suatu fungsi, dimana program hanya dijalankan sekali saja, sedangkan void loop () adalah suatu fungsi, dimana program akan dijalankan berulang kali hingga sistem dimatikan dayanya.

2.2.5 Kapasitor

Kapasitor atau Kondensator adalah suatu komponen elektronika pasif dimana memiliki fungsi untuk menyimpan muatan listrik sementara. Kapasitor menggunakan satuan Farad sebagai satuan kapasitansinya. Untuk digunakan pada ATmega328P, digunakanlah kapasitor 22pf (Piko Farad). Gambar Kapasitor ditunjukkan pada Gambar 2.5.



Gambar 2.4 Kapasitor 22pf
Sumber : (Purwanto, 2016)

2.2.6 Kristal Osilator

Kristal Osilator adalah komponen elektronika yang berfungsi untuk menghasilkan sinyal *pulse* yang relatif stabil dimana sinyal *pulse* sangat dibutuhkan perangkat elektronika lain yang memiliki *clock* khususnya ATmega328P. Pada ATmega328P, cukup menggunakan Kristal Osilator 16 MHz. Gambar 2.6 merupakan gambar untuk kristal osilator.

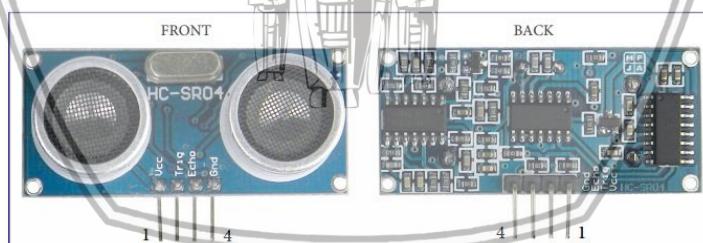


Gambar 2.5 Kristal Osilator

Sumber : (Kiswoyo, 2017)

2.2.7 Sensor Ultrasonik

Sensor ultrasonik HC-SR04 berfungsi sebagai sensor yang membaca jarak antara manusia dengan layar monitor. Sensor ini memiliki 4 pin dimana keempat pin tersebut adalah pin vcc, trig, echo, dan gnd. Selain 4 pin, sensor ini memiliki 2 bagian penting untuk membaca jarak, yaitu *transmitter* dan *receiver*. Cara kerja sensor ini adalah, pertama sensor akan mentransmit gelombang ultrasonik melalui *transmitter* lalu jika ada benda di depannya maka gelombang tersebut akan dipantulkan. Gelombang yang dipantulkan ini lah yang akan di tangkap oleh *receiver*. Gambar 2.7 merupakan gambar untuk sensor ultrasonik.



Gambar 2.6 Sensor Ultrasonik

Sumber : (depokinstruments, 2006)

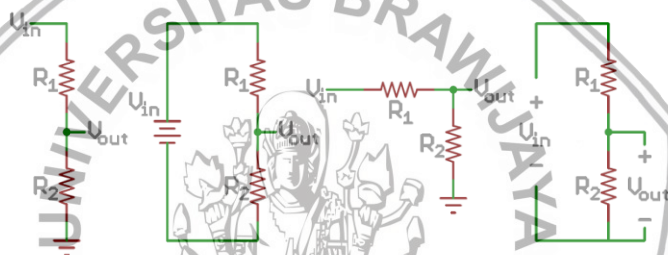
2.2.8 Sensor Cahaya LDR

Sensor Cahaya LDR (*Light Dependent Resistor*) adalah suatu resistor dimana resistansinya akan berubah tergantung dari keadaan cahaya disekitar. Resistansinya akan besar ketika sensor ini dipasang pada tempat yang gelap lalu resistansinya akan kecil jika dipasang pada tempat yang terang. Pemasangan sensor ini dapat diperlakukan sebagai resistor sebagaimana pada umumnya. Gambar 2.8 merupakan gambar sensor cahaya.



Gambar 2.7 Sensor Cahaya
Sumber : (zoniaelektro, 2014)

Sensor LDR hanya adalah sejenis resistor, nilai resistansi tidak dapat dibaca oleh ATmega328P oleh karena itu agar nilai resistansi yang diterima oleh sensor LDR dapat dibaca oleh ATmega328P, maka dibutuhkanlah rangkaian pembagi tegangan dimana fungsi dari rangkaian ini adalah agar nilai resistansi yang diterima oleh sensor LDR dapat dibaca oleh ATmega328P sebagai nilai tegangan, karena luaran dari rangkaian pembagi tegangan ini adalah voltase/tegangan. Ilustrasi dari rangkaian pembagi tegangan dapat dilihat pada gambar 2.9



Gambar 2.8 Rangkaian Pembagi Tegangan
Sumber : (Sparkfun, 2013)

Selain rangkaian di atas, terdapat persamaan untuk rangkaian pembagi tegangan, adapun persamaan nya dapat dilihat pada persamaan 2.1 di bawah :

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2} \quad (2.1)$$

Berdasarkan persamaan 2.1 terdapat beberapa keterangan yang harus diketahui, diantaranya sebagai berikut :

V_{out} = Voltase *Output* adalah voltase luaran yang dihasilkan yang nantinya akan menjadi nilai masukan sensor LDR ke ATmega328P

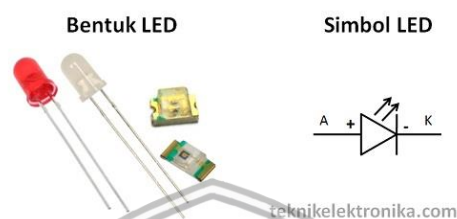
V_{in} = Voltase *Input* adalah voltase masukan dari *powerbank* yaitu sebesar 5V

R_2 = Resistor 2 yaitu resistor kedua dengan nilai 10k ohm

R_1 = Resistor 1 yaitu sensor LDR itu sendiri dengan nilai tergantung dari lingkungan

2.2.9 Lampu LED

Lampu LED atau lampu *Light Emitting Diode* adalah suatu jenis komponen elektronika yang akan memancarkan cahaya ketika diberikan sinyal *HIGH* dan akan mati jika diberikan sinyal *LOW*. LED merupakan suatu Dioda yang terbuat dari bahan semikonduktor. LED memiliki tegangan maksimal yang masuk sebesar 5volt lalu arus maksimal yang dapat diterima adalah 30mA (OptoSupply Inc., 2000). Mengacu pada gambar 2.10 merupakan gambar dari LED dan simbolnya.



Gambar 2.9 LED dan Simbol LED

Sumber : (Kho, 2018)

2.2.10 Meteran

Meteran merupakan alat ukur untuk mengukur panjang objek atau jarak. Untuk menggunakan meteran, sangat mudah karena penggunaan meteran hampir sama dengan penggunaan mistar/penggaris. Hanya yang membedakan adalah bahan pembuatannya. Secara umum, penggunaan meteran untuk mengukur jarak dan tinggi badan. Gambar 2.11 menunjukkan gambar dari meteran.



Gambar 2.10 Meteran

Sumber : (Kiswari, 2016)

2.2.11 Luxmeter

Mengacu pada gambar 2.12, luxmeter merupakan alat ukur untuk mengukur tingkat intensitas cahaya pada suatu tempat khususnya pada suatu ruangan. Satuan yang digunakan adalah Lux. Luxmeter bekerja secara otomatis untuk menghitung intensitas cahaya, sensor yang ada pada Luxmeter cukup sensitif dan linier. Semakin jauh jarak cahaya dengan luxmeter yang berarti cahaya sedang

redup maka luxmeter akan menunjukkan nilai yang kecil. Dan semakin dekat jarak cahaya dengan luxmeter maka luxmeter akan menunjukkan angka yang besar.



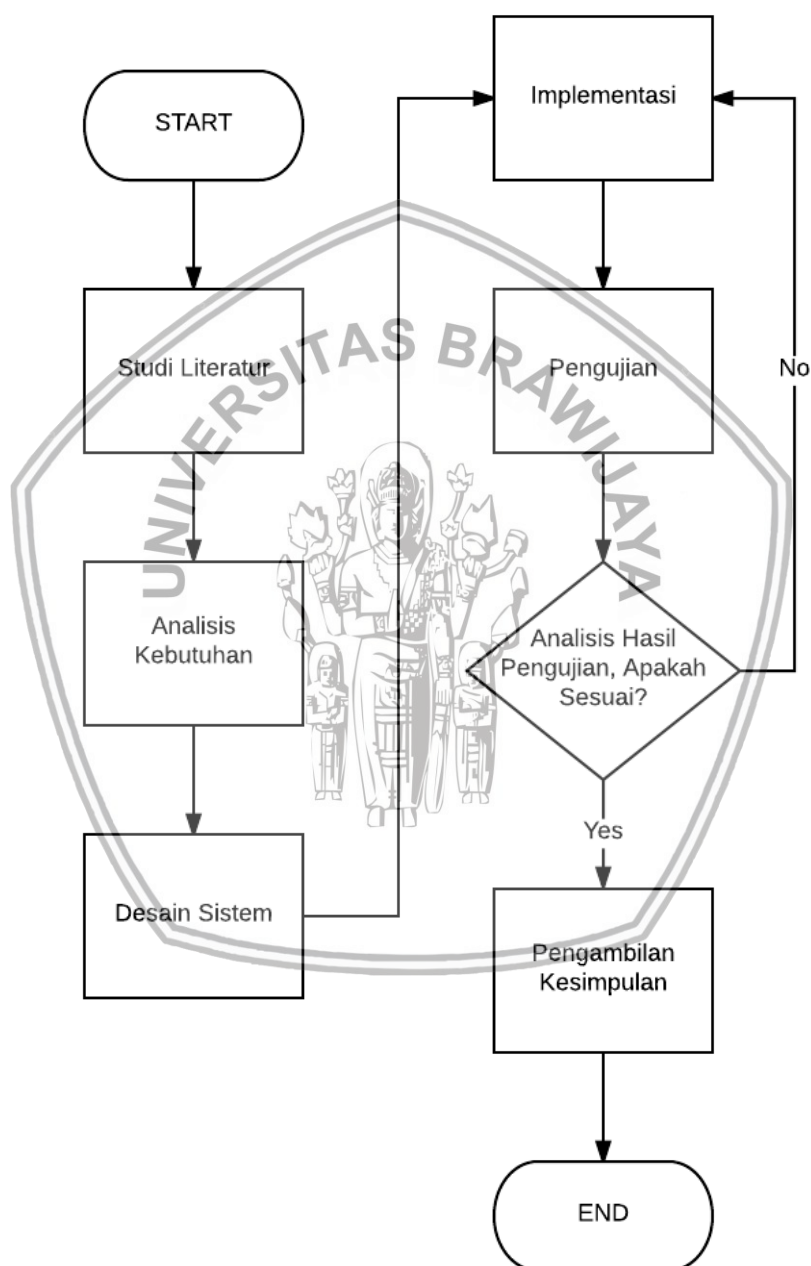
Gambar 2.11 Luxmeter

Sumber : (Azril, 2012)



BAB 3 METODOLOGI

Pada bagian ini dijelaskan tentang metode dan langkah-langkah yang dilakukan dalam pengerjaan penelitian ini. Penelitian bersifat implementatif. Berikut merupakan tahapan-tahapan metodologi penelitian yang diambil, tergambar pada diagram alir yang ditunjukkan gambar 3.1.



Gambar 3.1 Diagram Alir Alur Penelitian

Berdasarkan gambar 3.1, dapat diuraikan langkah-langkah metodologi penelitian yang akan dilakukan. Pertama akan dipelajari literatur-literatur yang

berhubungan dengan penelitian ini. Lalu akan dilakukan analisis kebutuhan sistem yang meliputi kebutuhan fungsional dan non-fungsional. Berikutnya akan didesain sistem agar sistem dapat bekerja dengan menggunakan konsumsi daya yang sedikit. Kemudian Implementasi sistem dengan menggunakan PCB yang berisi sensor-sensor serta Mikrokontroler ATmega328P. Selanjutnya pengujian sistem kepada manusia dimana *user* akan melakukan suatu kesalahan untuk men-tes apakah sistem berjalan dengan baik atau tidak, selain itu akan diuji juga bagaimana sistem dapat disebut sebagai sistem yang mengkonsumsi sedikit daya. Jika sudah diuji maka hasil dari pengujian tersebut akan dianalisis tingkat keberhasilannya, jika berhasil maka akan berlanjut ke penarikan kesimpulan jika gagal maka akan kembali ke implementasi.

3.1 Studi Literatur

Pada penelitian ini, dilakukan studi literatur untuk menjadi landasan dari desain dan implementasi sistem ini. Studi literatur mengenai bagaimana sistem dapat dikatakan sebagai sistem yang *low power* beserta bagaimana cara menimplementasikannya ke bahasa C++ atau ke Mikrokontroler ATmega328P. Studi literatur juga tentang bagaimana caranya agar suatu sensor dan parameter dapat menghasilkan suatu peringatan berdasarkan standar resmi. Selain itu dipelajari juga bagaimana cara kerja sensor ultrasonik, sensor cahaya, dan bagaimana mode *sleep* atau *wake* dapat terjadi. Selain cara kerja akan dipelajari juga *datasheet* dari alat-alat tersebut.

3.2 Analisis Kebutuhan

Analisis kebutuhan diperlukan untuk mengidentifikasi dan menganalisis apa-apa saja yang dibutuhkan agar dapat memenuhi kebutuhan penelitian ini. Kebutuhan analisis dibedakan dalam kebutuhan fungsional dan non-fungsional.

3.2.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah suatu kebutuhan tentang apa saja yang dapat dilakukan sistem, informasi apa yang didapat dari sistem, dan bagaimana hasil akhir dari sistem. Kebutuhan fungsional dari sistem sebagai berikut :

1. Sistem mendapatkan *input* berupa parameter-parameter yang sudah ditentukan.
2. Sistem menentukan apakah akan terjadi peringatan dengan menggunakan parameter-parameter berdasarkan standar resmi.
3. Sistem dapat mengeluarkan suatu luaran berdasarkan nilai yang dibaca.
4. Sistem dapat menentukan kapan sistem akan merubah ke mode *low power*.

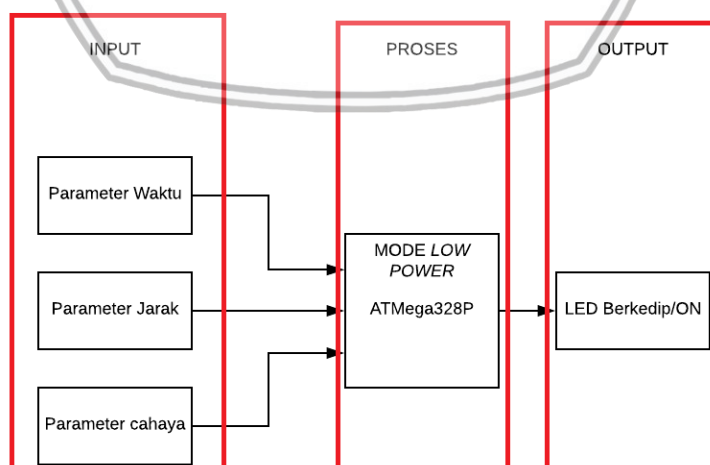
3.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan terhadap perangkat keras dan perangkat lunak yang digunakan untuk penelitian ini. Kebutuhan non-fungsional dari sistem adalah sebagai berikut :

1. Perangkat keras :
 - A. Laptop
 - B. PCB Buatan ukuran 6 cm x 7 cm (P x L)
 - C. Aklirik berukuran 8 cm x 8,5 cm x 5,5 cm (P x L x T)
 - D. ATmega328P
 - E. Sensor Ultrasonik HC-SR04
 - F. Sensor Cahaya LDR
 - G. LED
 - H. *Powerbank* Vivan Model M8 dengan kapasitas 8000mAh
 - I. Aklirik
2. Perangkat Lunak :
 - A. Windows 10
 - B. Arduino IDE 1.85
 - C. Library Event.h
 - D. Library Timer.h
 - E. Library LowPower.h

3.3 Desain Sistem

Desain sistem memuat langkah-langkah perencanaan suatu sistem agar sistem dapat berjalan sesuai dengan analisis kebutuhan. Untuk memenuhi kebutuhan sistem yaitu menghasilkan sistem yang dapat memberikan notifikasi kepada pengguna namun ditanamam juga fungsi agar sistem tersebut *low power*, langkah-langkahnya dapat dilihat dari gambar 3.2 di bawah :



Gambar 3.2 Diagram Blok Sistem

Berdasarkan gambar 3.2 terdapat 3 tahapan pada perancangan sistem yang dapat dijelaskan sebagai berikut :

1. Ketiga parameter yaitu waktu (oleh ATmega328P), jarak (dibaca oleh sensor ultrasonik) dan cahaya (dibaca oleh sensor cahaya) akan menjadi *input* sistem.
2. Ketiga parameter tersebut menjadi *input* ATmega328P. Jika sistem dalam keadaan aktif maka akan diproses apakah akan ada notifikasi peringatan kepada pengguna. Jika sistem dalam keadaan tidak digunakan maka sistem akan menjadi *low power*, jika tidak ada interupsi maka sistem akan *sleep* namun jika ada interupsi maka sistem akan *wake*.
3. *Output* akan ditentukan berdasarkan keadaan sistem. Jika ada peringatan maka LED akan berkedip, dan jika tidak ada peringatan maka LED akan menyala.

3.4 Implementasi

Implementasi berisikan beberapa langkah yang diambil agar sistem dapat terwujud sesuai dengan desain sistem atau perancangan sistem. Berdasarkan perancangan sistem, implementasi dapat meliputi hal-hal berikut :

1. Instalasi perangkat lunak seperti Arduino IDE .
2. Merancang sistem berdasarkan *software*-nya dengan Arduino IDE, dimana isinya adalah program tentang jarak, waktu, dan intensitas cahaya serta program ketika sistem akan *sleep* dan *wake*.
3. Merancang sistem berdasarkan *hardware*-nya dimana *hardware* ini terdiri dari 1 Sensor Ultrasonik, 1 Sensor Cahaya, 3 LED, 1 Mikrokontroler ATmega 328P, 2 Kapsitor 22 pf dan 1 Crystal Osilator. Sensor Ultrasonik dan sensor cahaya menjadi *hardware* masukan untuk ATmega328P, lalu LED akan menjadi luaran, Kapasitor dan Kristal Osilator akan menjadi perangkat yang akan membantu ATmega328P bekerja. Serta dirancang bentuk *prototype* dari sistem.

3.5 Pengujian dan Analisis

Pengujian dan analisis sistem dilakukan untuk menguji tingkat keberhasilan sistem dalam menentukan peringatan kepada pengguna berdasarkan masukan-masukan yang ada, pengujian-pengujiannya adalah :

1. Pengujian pertama dilakukan untuk menguji akurasi sensor dimana peralatan pembantunya adalah meteran dan Luxmeter. Serta menguji fungsionalitas *timer* pada sistem.
2. Pengujian kedua adalah menguji arus pada saat *low power*, ada berapa jumlah arus yang digunakan ketika sistem dalam keadaan *sleep* atau *wake*.
3. Pengujian ketiga adalah menguji waktu *wake* pada sistem dalam satuan milisekon.

3.6 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan berdasarkan analisis dan pengujian terhadap sistem yang sudah berhasil menentukan luaran yang sesuai. Berdasarkan hasil pengujian, maka dapat ditentukan luaran yang tepat yang akan dijalankan sistem sesuai dengan keadaan sistem tersebut, apakah sistem tersebut sedang tidak digunakan atau sedang aktif. Jika luaran yang diharapkan dan semestinya belum sesuai maka akan dilakukan kembali tahapan implementasi untuk meninjau ulang keadaan sistem baik keadaan *hardware* maupun *software*.



BAB 4 ANALISIS KEBUTUHAN

Pada bab 4 analisis kebutuhan terdapat sub bab deskripsi sistem dimana deskripsi sistem terdiri dari sub bab lagi. Inti dari deskripsi sistem adalah membahas mengenai sistem yang diteliti secara umum. Sub bab berikutnya adalah kebutuhan sistem yang memiliki inti yang membahas mengenai kebutuhan yang dibutuhkan sistem agar sistem dapat bekerja sesuai dengan prinsipnya.

4.1 Deskripsi Umum Sistem

Deskripsi umum merupakan gambaran secara umum seperti apa sistem yang sedang peneliti rancang. Sistem yang peneliti rancang adalah suatu alat yang akan memberikan peringatan kepada orang yang bermain *game* secara berlebihan, dimana peringatan tersebut dengan sebuah lampu LED yang berkedip. Sistem terdiri dari Mikrokontroler ATmega328P, Sensor Ultrasonik, Sensor Cahaya, dan *powerbank* sebagai sumber daya tegangannya.

Sistem ini akan ditempatkan di atas layar monitor dengan cara menjepitnya, oleh karena ini sistem ini juga bersifat *portable*, artinya sistem dapat ditempatkan di layar monitor mana saja sesuai dengan kebutuhan pengguna. Selain itu, sistem juga dirancang sedemikian rupa dengan menggunakan PCB buatan sehingga tidak diperlukan lagi penggunaan *project board*. Oleh karena itu, akan dibuatkan suatu *case* dari mika sedemikian rupa agar *case* tersebut dapat diisi oleh PCB buatan dan juga alat penjepit untuk monitor.

Karena sistem ini memberikan notifikasi saja dengan sumber daya tegangannya berupa *powerbank*, maka sistem ini akan diimplementasikan metode *low power* yang berfungsi untuk menghemat daya *powerbank* ketika sistem sedang tidak digunakan. Ketika sistem tidak digunakan maka sistem akan memasuki mode *low power* yang selanjutnya sistem akan *sleep*. Ketika sistem *sleep* maka sistem akan menonaktifkan beberapa fungsi dari ATmega328P dan ini akan membuat penggunaan sumber daya *powerbank* lebih efisien daripada ketika digunakan. Jika sistem sedang *sleep* maka sistem juga harus bisa kembali aktif atau *wake*. Sistem akan *wake* ketika terjadi *external interrupt* yang dilakukan oleh pengguna. *External Interrupt* terjadi ketika pengguna menutup sensor cahaya sehingga nilai pada sensor cahaya akan mendekati 0. Ketika sistem sudah *wake* maka sistem akan menjalankan fungsionalitasnya seperti semula.

4.1.1 Karakteristik Pengguna

Pengguna ditujukan sebagai *single user* artinya penggunanya adalah orang yang sedang menggunakan komputer terutama menggunakan komputer untuk bermain *game*. Pengguna ini akan duduk di depan monitor dimana di atas monitor terdapat sensor ultrasonik dan sensor LDR. Jarak antara pengguna dengan sensor ultrasonik akan diukur terus menerus oleh sistem dan sistem akan mengeluarkan luaran berupa LED yang berkedip jika pengguna terlalu dekat

dengan monitor, begitu juga dengan pencahayaan, LED juga akan berkedip jika ruangan gelap dan sensor LDR membaca nilai yang cukup kecil. Ketika pengguna pergi keluar atau pengguna sedang tidak ada ditempat dan keadaan sistem sedang menyala, sistem akan masuk ke mode *low power sleep powerdown*. Untuk menggunakan sistem lagi, pengguna hanya cukup menutup sensor LDR, maka sistem akan masuk ke mode *wake* dengan *external interrupt*.

4.1.2 Ruang Lingkup Sistem

Ruang lingkup sistem atau lingkungan sistem yang berjudul "Implementasi *Low Power* Pada Sistem Notifikasi Untuk *Gamer* Yang Bermain *Intense* Berbasis *Embedded System*" adalah :

1. Sistem di tempatkan di atas monitor dengan cara menjepit layar monitor tersebut .
2. Sistem menggunakan *powerbank* sebagai sumber daya-nya.
3. Sistem akan dimuat pada sebuah PCB berukuran 7 cm x 6 cm (P x L) dimana komponen-komponennya adalah sensor ultrasonik, sensor LDR, mikrokontroler ATmega328P, Kristal Osilator, Kapasitor, dan tempat menaruh *powerbank* lalu dikemas dengan akrilik berukuran 14 x 8 x 5,5 cm (P x L x T).
4. Sistem di tempatkan pada layar monitor yang tidak berantakan sehingga tidak ada apapun yang menghalangi sensor bekerja .

4.1.3 Batasan Sistem

Dalam perancangan sistem, terdapat batasan-batasan dalam implementasi sistem. Batasan-batasan tersebut meliputi :

1. Pemanfaatan *low power* diimplementasikan ke IC ATmega328P dimana IC tersebut akan dimuat pada PCB buatan .
2. Sensor ultrasonik, sensor LDR, serta LED juga dimuat ke PCB buatan tersebut
3. Digunakan akrilik untuk membuat *case* dari sistem.
4. Sistem membaca nilai jarak antara pengguna dengan monitor dengan Sensor Ultrasonik .
5. Sistem membaca nilai intensitas cahaya dalam satuan lux dengan Sensor LDR .
6. Sistem diimplementasikan *low power* untuk pengguna komputer khususnya *gamer* .
7. Luaran dari sistem adalah lampu LED yang menyala atau berkedip.
8. Sistem diletakkan di atas layar monitor dimana sistem tersebut bersifat *portable* artinya dapat dipindah-pindah.
9. Sistem digunakan pada ruangan tertutup. Karena jika digunakan pada ruangan terbuka, akan mempengaruhi pencahayaannya. Jika terlalu terang maka layar monitor akan tidak terlihat, lalu jika terlalu gelap maka dapat membuat pengguna merasa pusing.

4.2 Kebutuhan Sistem

Kebutuhan Sistem meliputi apa-apa saja yang dibutuhkan sistem. Kebutuhan sistem meliputi Kebutuhan Fungsional dan Kebutuhan Non-Fungsional dimana di dalamnya terdapat Kebutuhan Perangkat Keras dan kebutuhan Perangkat Lunak.

4.2.1 Kebutuhan Fungsional

Kebutuhan Fungsional adalah kebutuhan-kebutuhan sistem agar sistem dapat berjalan sesuai dengan yang diharapkan. Kebutuhan fungsional antara lain :

1. Sistem mendapatkan *input* berupa parameter-parameter yang sudah ditentukan.

Parameter-parameter yang akan menjadi *inputan* kepada sistem yang berjudul “Implementasi *Low Power* Pada Sistem Notifikasi Untuk *Gamer* Yang Bermain *Intense* Berbasis *Embedded System*” adalah parameter jarak dalam satuan centimeter (cm) dimana parameter jarak didapat dari sensor ultrasonik, parameter intensitas cahaya dalam satuan lux dimana parameter intensitas cahaya didapat dari sensor *Light Dependent System* (LDR) lalu parameter waktu dalam satuan detik (s) dimana parameter waktu didapat dari mikrokontroler ATmega328P.

2. Sistem menentukan apakah akan terjadi peringatan dengan menggunakan parameter-parameter berdasarkan standar resmi.

Peringatan yang akan dihasilkan sistem yang berjudul “Implementasi *Low Power* Pada Sistem Notifikasi Untuk *Gamer* Yang Bermain *Intense* Berbasis *Embedded System*” adalah peringatan berupa lampu dari *Light Emitting Diode* yang menyala ketika pengguna melewati batasan-batasan dari parameter sistem. Untuk batasan jarak, berdasarkan standar dari *Occupational Safety and Health Administration* (OSHA) Amerika Serikat, jarak minimal antara pengguna dan layar monitor adalah 45 cm (OSHA, 1997). Untuk batasan intensitas cahaya, berdasarkan Keputusan Menteri Kesehatan Republik Indonesia Nomor 1405/MENKES/SK/XI/2002 tentang Persyaratan Kesehatan Lingkungan Kerja Perkantoran Dan Industri, pencahayaan di ruangan minimal 100 Lux (Kepmenkes RI, 2002). Untuk batasan waktu, waktu maksimal dalam pengguna menggunakan komputer adalah 4 jam, lebih dari itu maka akan berbahaya bagi pengguna (Ploeg, 2012).

3. Sistem dapat mengeluarkan suatu luaran berdasarkan nilai yang dibaca.

Dengan mengacu terhadap batasan-batasan dari parameter sistem yang telah disebutkan pada poin nomor 2, maka jika pengguna, misalnya, berjarak 40 cm dari monitor maka lampu LED akan berkedip. Jika pengguna menggunakan komputer pada ruangan dengan intensitas pencahayaannya kurang dari 100 maka lampu LED akan berkedip. Dan jika pengguna telah menggunakan komputer lebih dari 4 jam, maka lampu LED akan berkedip.

4. Sistem dapat menentukan kapan sistem akan merubah ke mode *Low Power*.

Sistem yang berjudul “Implementasi *Low Power* Pada Sistem Notifikasi Untuk *Gamer* Yang Bermain *Intense* Berbasis *Embedded System*” dapat merubah ke mode *Low power* dimana akan aktif ketika pengguna berada pada jarak di atas 100 cm dari layar monitor. Ini mengindikasikan bahwa pengguna sedang melakukan aktifitas lain namun masih menghidupkan sistem. Tujuan dari sistem menjadi *Low Power* adalah untuk menghemat daya yang digunakan ketika tidak sedang digunakan. Sistem akan aktif kembali ketika pengguna menutup sensor LDR yang akan *men-trigger external interrupt* dan akan menghidupkan kembali sistemnya.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan-kebutuhan yang akan mendukung kinerja sistem. Kebutuhan non-fungsional dibedakan menjadi kebutuhan perangkat keras dan kebutuhan perangkat lunak :

1. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras meliputi laptop, ATmega328P, sensor ultrasonik hc-sr04, sensor cahaya ldr, led, *powerbank*, PCB buatan dan multimeter. Penjelasan fungsi tiap-tiap komponen dapat dilihat pada tabel 4.1.

Tabel 4.1 Penjelasan fungsi perangkat keras sistem

Perangkat Keras	Fungsi
Laptop	Digunakan untuk menunggah kode program melalui program IDE dan untuk memonitoring data pada serial monitor
Mikrokontroler ATmega328P	Sebagai kontroler untuk sistem dimana mikrokontroler ini akan mengolah data-data yang didapat dari parameter-parameter dan menampilkan suatu luaran jika memang ada. lalu mengolah juga program mengenai <i>Sleep</i> dan <i>Wake</i>
Sensor Ultrasonik HC-SR04	Sebagai sensor untuk membaca jarak, dimana nilai yang dibaca oleh sensor akan dikirim ke mikrokontroler ATmega328P untuk diproses

Tabel 4.1 (Lanjutan)

Perangkat Keras	Fungsi
Sensor <i>Ligt Dependent Resistor</i> (LDR)	Sebagai sensor untuk membaca kadar intensitas cahaya, dimana nilai yang dibaca oleh sensor akan dikirim ke mikrokontroler ATmega328P untuk diproses
<i>Light Emitting Diode</i> (LED)	Sebagai indikator untuk luaran sesuai yang dibaca oleh ketiga parameter
PCB Buatan	Sebagai tempat untuk Mikrokontoler ATmega328P, Sensor Ultrasonik, Sensor LDR, LED dan <i>powerbank</i> saling berkomunikasi satu sama lain
Aklirik	Sebagai kemasan untuk PCB buatan, selain itu aklirik nantinya dengan bantuan penjepit akan menjadikan sistem <i>portable</i>
<i>Powerbank</i>	Sebagai sumber daya tegangan agar sistem dapat aktif
Multimeter	Sebagai alat ukur sistem, seperti mengukur tegangan, mengukur resistansi, dan mengukur arus

2. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak adalah kebutuhan yang berhubungan dengan program atau perangkat lunak lainnya. Kebutuhan perangkat lunak meliputi Windows 10, IDE, ultrasonik.h, ldr.h, waktu.h, Library Low Power dimana fungsi dari perangkat lunak tersebut dapat dilihat pada tabel 4.2.

Tabel 4.2 Fungsi-Fungsi Perangkat Lunak

Perangkat Lunak	Fungsi
Windows 10	Sebagai sistem operasi pada perangkat keras laptop, dimana digunakan untuk membuat program, membuat skematik, dan membuat diagram
IDE	Sebagai perangkat lunak untuk memprogram ATmega328P. Selain itu juga digunakan untuk memprogram parameter-parameter dan Low Power

Tabel 4.2 (Lanjutan)

Perangkat Lunak	Fungsi
Event.h	Sebagai library untuk mempermudah memprogram <i>timer</i> pada ATmega328P
Timer.h	Sebagai library untuk mempermudah memprogram <i>timer</i> pada ATmega328P
lowpower.h	Sebagai library untuk mempermudah memprogram Low Power pada ATmega328P

4.2.3 Kebutuhan Peformansi Sistem

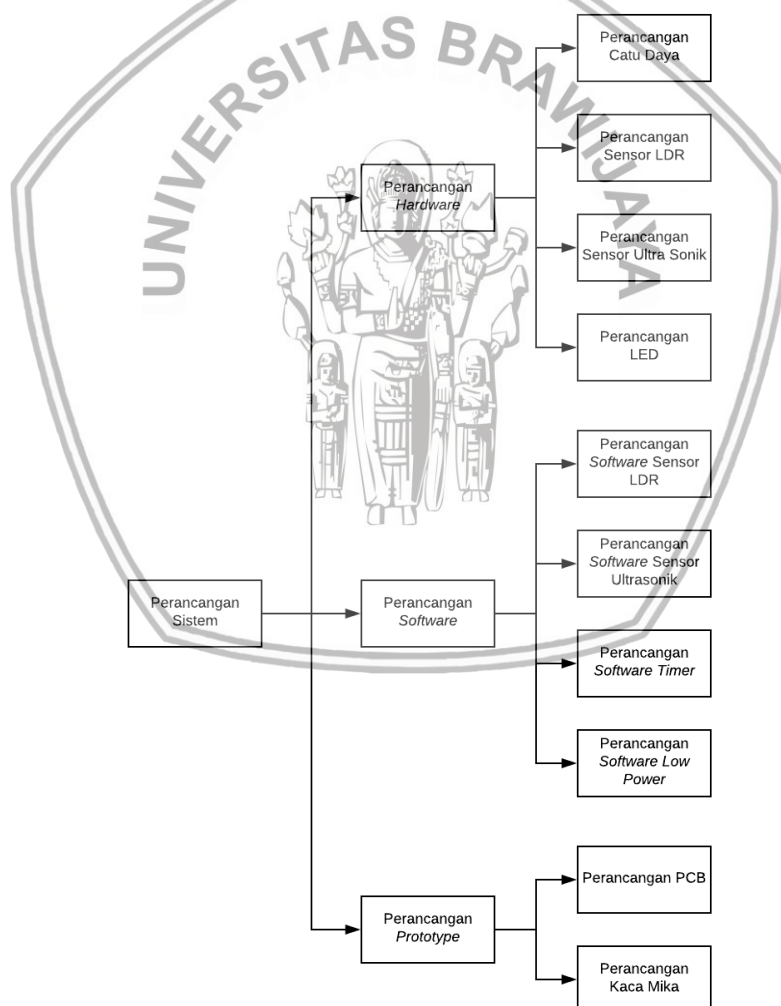
Kebutuhan peformansi sistem adalah segala kebutuhan agar sistem dapat bekerja secara maksimal. Sistem dapat berjalan dengan semestinya ketika segala faktor-faktor pendukung sistem seperti kebutuhan fungsional dan non-fungsional dapat terpenuhi. Seperti halnya sensor harus dapat membaca nilai dengan tingkat kesalahan seminimal mungkin dan sistem dapat menginisialisasikan mode *sleep* atau *wake*. Sistem juga harus berada dilingkungan rapi dan tidak berantakan, karena kerja sistem sangat bergantung dengan faktor lingkungan disekitar. Misalnya ketika sistem digunakan pada ruangan terbuka, ketika pencahayaan di luar sangat terang sistem tidak mengkedipkan lampu LED yang berarti *gamer* bermain dengan normal, namun cahaya yang berlebihan dapat membuat layar monitor terlihat redup sehingga ini akan membuat mata pengguna terasa sakit karena dipaksa untuk melihat layar monitor yang redup. Sistem diharapkan dapat berjalan sesuai yang diharapkan peneliti.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab perancangan dan implementasi akan dijabarkan mengenai bagaimana sistem dirancang atau dibuat baik secara *hardware* maupun *software* serta bagaimana sistem diimplementasikan metodenya. Sistem juga akan dirancang ke dalam bentuk *prototype* sehingga sistem dapat berjalan sesuai dengan prinsip kerjanya.

5.1 Perancangan Sistem

Pada tahapan perancangan sistem akan dibahas mengenai tahapan-tahapan yang ada pada sistem, dimulai dari perancangan *hardware*, perancangan *prototype*, lalu perancangan *software* dimana tahapan perancangan dapat dilihat pada gambar 5.1.



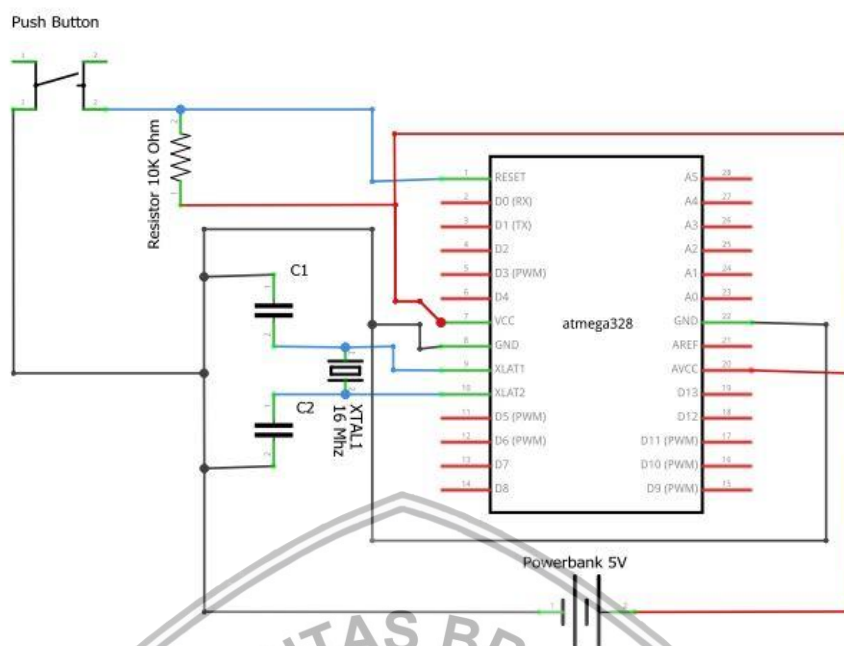
Gambar 5.1 Diagram Perancangan Sistem

Tabel 5. 1 Keterangan Pin Perancangan *Hardware*

ATmega32 8P	PB	XTAL	Butto n	S.LD R	S.Ultr asonik	LED 1	LED 2	LED 3	Ket.
1			+						Reset
4				-					D2
7	+			+	VCC				VCC
8	-		-		GND				GND
9		+							XTAL 1
10		+							XTAL 2
11					ECHO				D5
12					TRIG				D6
14						+			D8
15							+		D9
16								+	D10
20	+								AVCC
22	-								GND
23									A0

5.1.1.1 Perancangan Catu Daya

Perancangan Catu Daya pada sistem menggunakan *Powerbank* Vivan model M8 dengan kapasitas 8000 mAh. *Powerbank* dipilih sebagai sumber tegangan utama sistem karena *powerbank* dinilai dapat meminimalisir energi yang dihasilkan sehingga akan hemat energi. *Powerbank* juga merupakan salah satu sumber tegangan yang dapat dengan mudah ditemukan serta memiliki arus dan voltase yang relatif stabil. Untuk perancangannya, kutub positif dan kutub negatif kabel USB disambungkan ke pin VCC dan GND mikrokontroler ATmega328P, untuk menyalakan sistem cukup menyambungkan kabel USB ke *powerbank*. Selain *powerbank*, terdapat *push button* yang disambungkan pada reset yang berfungsi untuk mereset sistem. Pada pin reset disambung juga dengan resistor 10K ohm dimana berfungsi untuk menahan tegangan yang masuk ke pin reset agar tidak terlalu besar. Adapun skematik dari perancangan catu daya dapat dilihat pada gambar 5.3 dan keterangan pin ditunjukkan pada tabel 5.2.



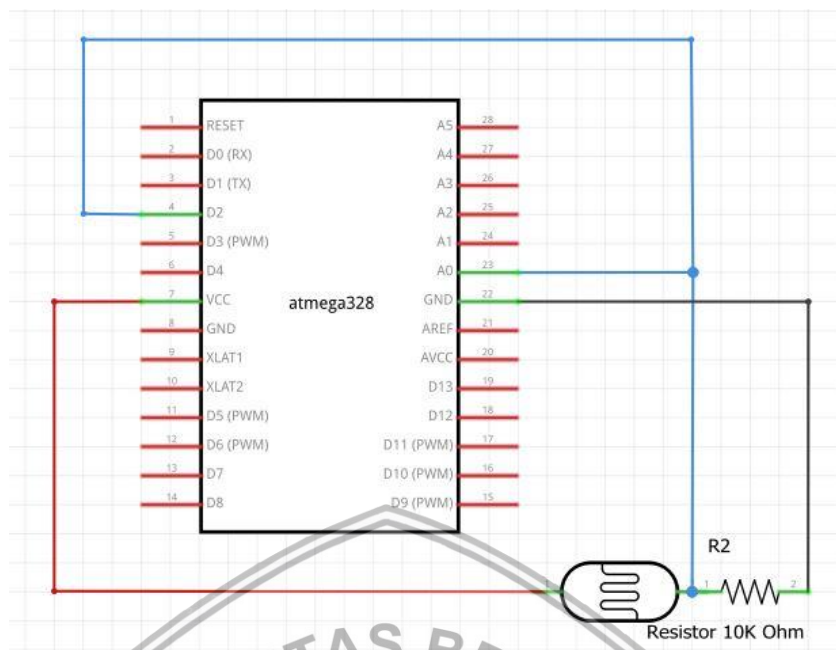
Gambar 5.3 Skematik Perancangan Catu Daya

Tabel 5. 2 Keterangan Pin Perancangan Catu Daya

Powerbank	Pin ATmega328P	Keterangan
Kutub Positif (+)	7 dan 20	VCC dan AVCC
Kutub Negatif (-)	8 dan 22	GND

5.1.1.2 Perancangan Sensor LDR

Perancangan Sensor LDR dimulai dengan menyambungkan VCC pada pin ATmega328P kepada sensor LDR, lalu sensor LDR disambungkan ke pin A0 dimana A0 sendiri sebagai pembaca nilai analog yang dihasilkan sensor LDR. Lalu juga disambungkan ke pin D2 dimana pin D2 sebagai pin yang akan *interrupt* ketika sistem dalam keadaan *sleep*, dengan kata lain, pin D2 akan menjadi pemicu *wake* ketika sistem sedang *sleep*. Sensor LDR juga harus dirangkai dengan rangkaian pembagi tegangan yang berfungsi untuk membagi tegangan yang masuk ke pin A0 dan juga membaca nilai dari sensor LDR itu sendiri. Penggunaan resistor 10k ohm bertujuan agar perhitungan sensor LDR untuk menghasilkan nilai lux dapat setara/mendekati dengan yang dihasilkan oleh luxmeter. Gambar skematik dari perancangan sensor LDR dilihat pada gambar 5.4 dan tabel 5.3 merupakan keterangan pin-pin yang digunakan.



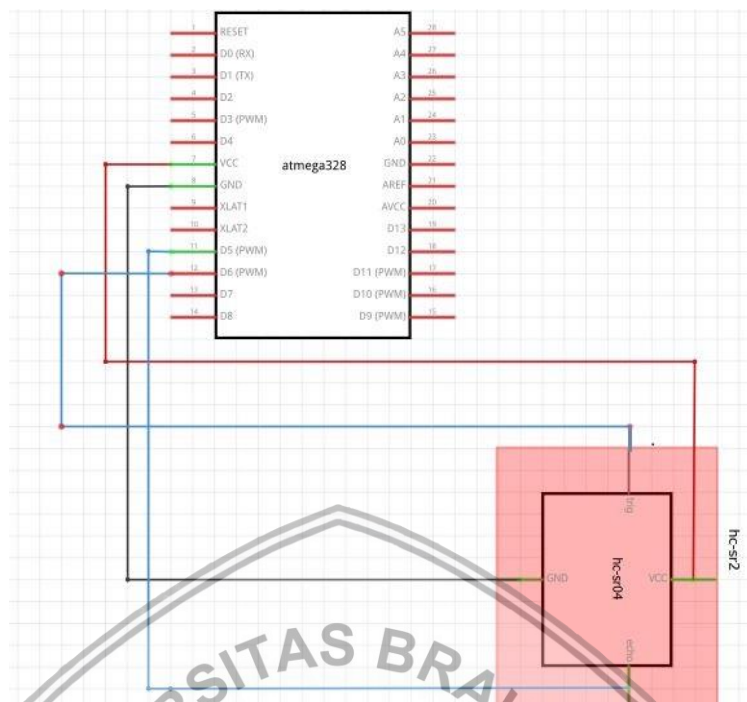
Gambar 5.4 Skematik Perancangan Sensor LDR

Tabel 5. 3 Keterangan Pin Perancangan Sensor LDR

LDR	Pin ATmega328P	Keterangan
Kutub Positif (+)	7	VCC
Kutub Negatif (-)	22,23,4	GND,A0,D2

5.1.1.3 Perancangan Sensor Ultrasonik

Perancangan sensor ultrasonik dimulai dengan menyambungkan VCC pada ATmega328P kepada VCC sensor ultrasonik, begitupula dengan GND pada ATmega328P yang disambungkan ke GND sensor ultrasonik. Untuk pin Trig pada sensor ultrasonik disambungkan pada digital pin D6 lalu pin Echo pada sensor ultrasonik disambungkan ke digital pin D5. Perancangan sensor ultrasonik juga berguna untuk mendapatkan nilai masukan jarak antara sistem dengan pengguna dengan menghitung berapa lama waktu yang dibutuhkan gelombang ultrasonik yang ditembak-kan dari *trigger* kembali ke *echo* setelah gelombang tersebut dipantulkan jika ada benda di depannya. Waktu yang dibutuhkan akan menjadi indikator jarak dalam *centimeter*. Sensor ultrasonik juga menjadi pemicu *sleep* dari sistem dimana sistem akan *sleep* jika nilai yang diterima melebihi batas yang telah ditentukan. Skematik dari perancangan sensor ultrasonik dapat dilihat pada gambar 5.5. Tabel 5.4 merupakan keterangan pin-pin yang digunakan.



Gambar 5.5 Skematik Perancangan Sensor Ultrasonik

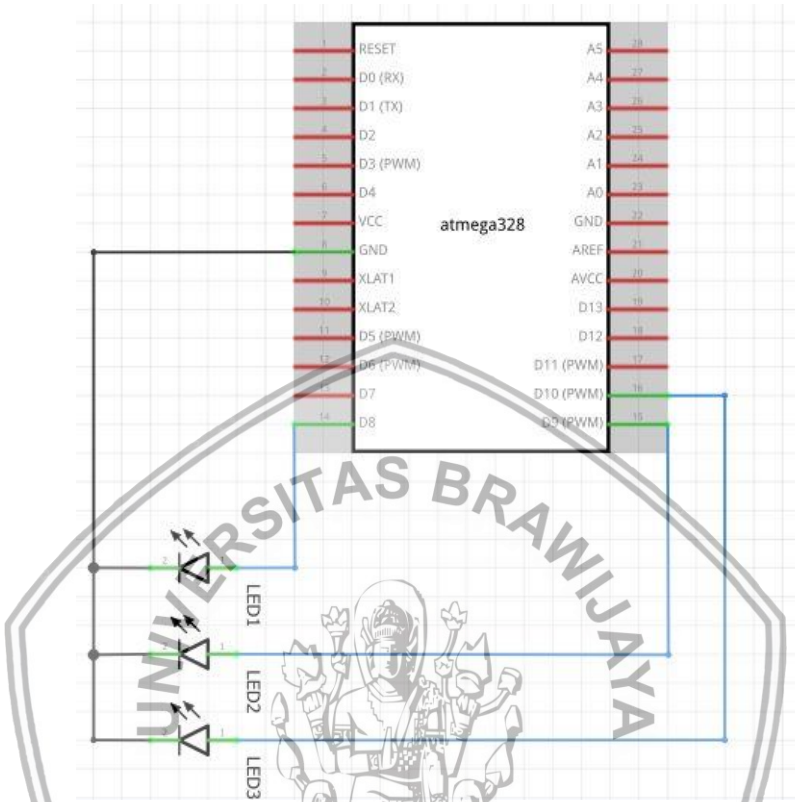
Tabel 5.4 Keterangan Pin Perancangan Sensor Ultrasonik

Ultrasonik	Pin ATmega328P	Keterangan
VCC	7	VCC
Trig	12	D6
Echo	11	D5
GND	8	GND

5.1.1.4 Perancangan LED

Perancangan LED berfungsi sebagai indikator luaran dari sistem. Tersedia 3 LED dimana LED 1 disambungkan pada pin D8 sebagai indikator LED untuk sensor LDR, LED 2 disambungkan ke pin D9 sebagai indikator LED untuk sensor ultrasonik, dan LED 3 disambungkan ke pin D10 sebagai indikator *timer*. LED akan berkedip jika pengguna menggunakan komputer melewati batas yang telah ditentukan. Misalnya LED 1 akan berkedip jika pengguna menggunakan komputer dalam ruangan yang gelap, atau LED 2 akan berkedip jika pengguna menggunakan komputer dengan jarak antara pengguna dan monitor sangat dekat. Pada perancangan LED tidak dibutuhkan resistor karena selain dapat membuat lampu LED redup, arus yang masuk ke lampu LED belum mencapai arus

maksimal yang dapat diterima oleh lampu LED, yaitu sebesar 18mA untuk arus yang masuk sedangkan arus maksimalnya 30mA . Skematik Perancangan LED dapat dilihat pada gambar 5.6 lalu keterangan pin pada tabel 5.5.



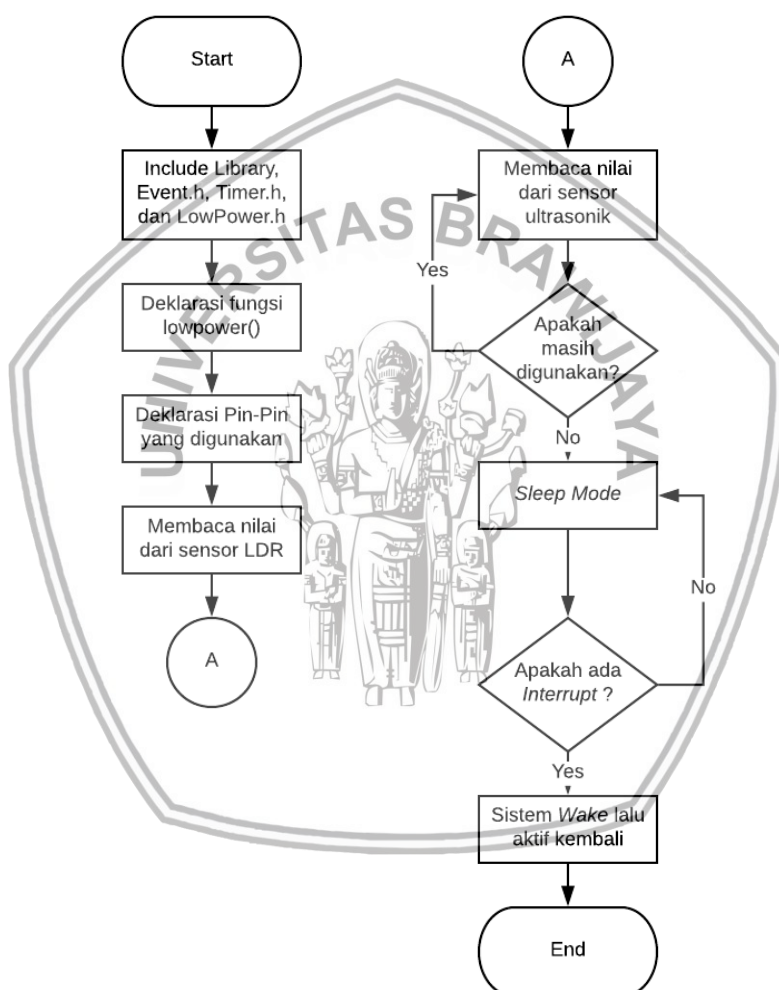
Gambar 5.6 Skematik Perancangan LED

Tabel 5.5 Keterangan Pin Perancangan LED

LED	Pin ATmega328P	Keterangan
Kutub Positif (+)	14,15,16	D8,D9,D10
Kutub Negatif (-)	8	GND

5.1.2 Perancangan *Software*

Perancangan *software* merupakan rangkaian alur kerja sistem dalam bentuk perangkat lunak atau *software*. Rangkaian perancangan *software* tersebut meliputi rangkaian inisialisasi *library* yang digunakan dalam sistem, lalu pembacaan nilai ADC oleh sensor LDR dalam satuan Lux, pembacaan nilai jarak dalam *centimeter* oleh sensor ultrasonik, serta rancangan perangkat lunak untuk *low power sleep power down*. Adapun perancangan *software* sistem dapat dilihat pada gambar 5.7.



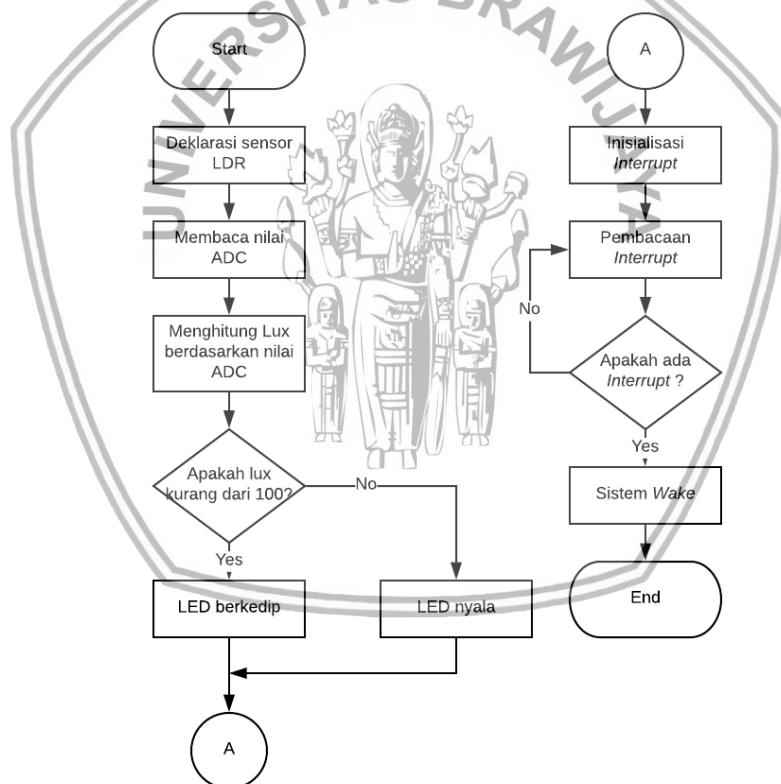
Gambar 5.7 Flowchart Perancangan *Software* Sistem

Pertama, kode program akan membaca *library* yang telah dideklarasikan seperti *library* event, timer dan *lowpower*, selanjutnya dibuat fungsi *lowpower* yang nanti akan digunakan sebagai fungsi yang dapat dipanggil oleh sensor ultrasonik. Pin-pin juga akan dideklarasikan sebelumnya agar dikenali oleh fungsi-fungsi lain. Lalu kode program akan membaca dan menghitung nilai lux berdasarkan jumlah kadar voltase yang masuk ke ADC. Selanjutnya sistem akan

membaca nilai sensor ultrasonik dimana luaran dari sensor ultrasonik akan menentukan sistem akan menjadi *lowpower* atau tidak. Ketika masuk ke mode *sleep* maka sistem dapat diwake dengan sensor LDR.

5.1.2.1 Perancangan *Software* Sensor LDR

Perancangan *software* dengan sensor LDR berfungsi untuk membaca kadar intensitas cahaya dalam satuan Lux. Kadar intensitas cahaya yang dibaca akan menentukan luaran dari sistem, ketika kadar lux mencapai kurang dari 100 lux, itu berarti ruangan cukup gelap, maka luaran berupa LED yang berkedip. Sensor LDR selain membaca lux dan memberikan luaran juga berperan dalam meng-*interrupt* sistem ketika sistem dalam keadaan *sleep*. Caranya hanya dengan menutup sensor LDR dimana akan memberikan sinyal *LOW* pada ATmega328P, maka sistem akan kembali aktif. Pada gambar 5.8 merupakan aliran sistem pada sensor LDR.



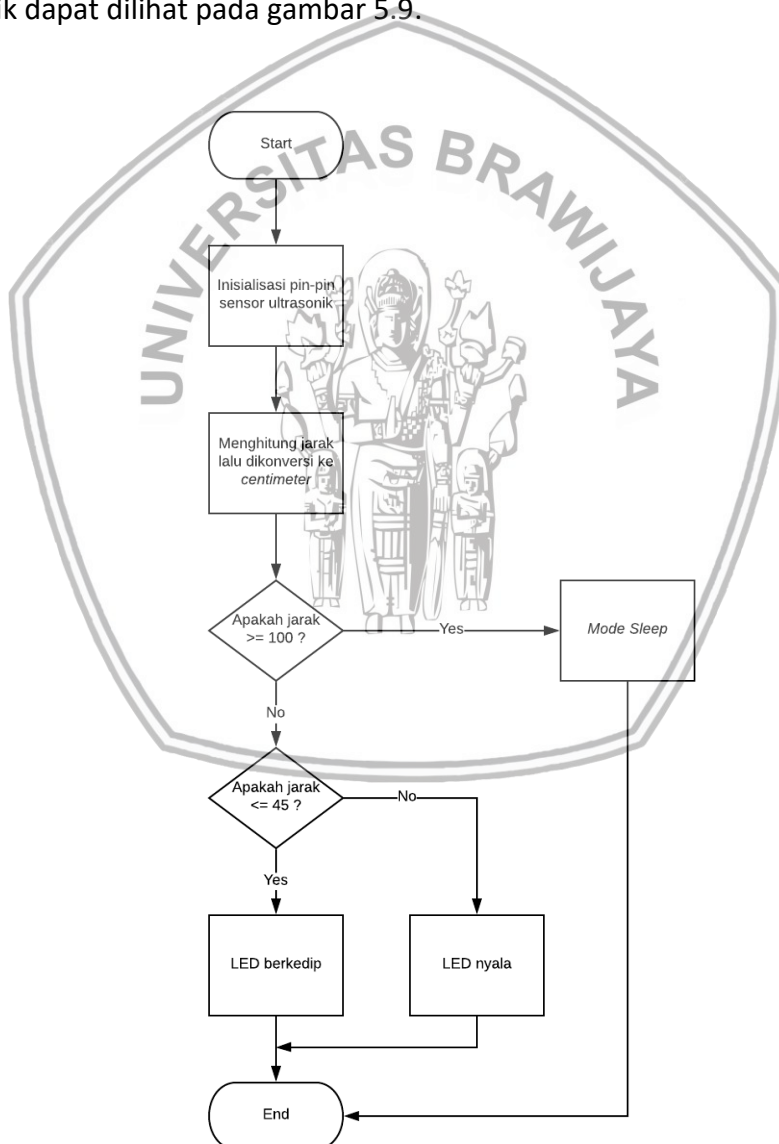
Gambar 5.8 Flowchart Perancangan Sensor LDR

Pin-pin dideklarasikan pada kode program yang selanjutnya akan dihasilkan nilai ADC dimana nilai ADC tersebut akan menjadi nilai intensitas cahaya dalam satuan lux. Lalu jika lux yang terdeteksi kurang dari 100 lux maka LED akan hidup sebaliknya akan berkedip. Untuk interupsi sensor LDR dapat menginterupsi sistem dalam keadaan *sleep* ke keadaan *wake* dengan cara memberikan *interrupt*

external berupa menurutnya kadar lux yang dibaca oleh sensor LDR, dengan begitu sistem akan *wake* dan berfungsi normal.

5.1.2.2 Perancangan *Software* Sensor Ultrasonik

Perancangan *software* sensor ultrasonik berfungsi untuk membaca nilai jarak antara monitor dengan pengguna dengan satuan *centimeter*. Nilai jarak yang dibaca oleh sensor ultrasonik akan menentukan luaran dari sistem. Jika nilai jarak yang dibaca oleh sensor ultrasonik kurang dari 45 cm, maka LED akan berkedip. Namun jika sensor ultrasonik membaca nilai jarak melebihi 100 cm, maka sistem akan masuk ke mode *low power sleep power down*, dimana untuk mengaktifkan sistem kembali dibutuhkan *interrupt* oleh sensor LDR. Diagram alir sensor ultrasonik dapat dilihat pada gambar 5.9.

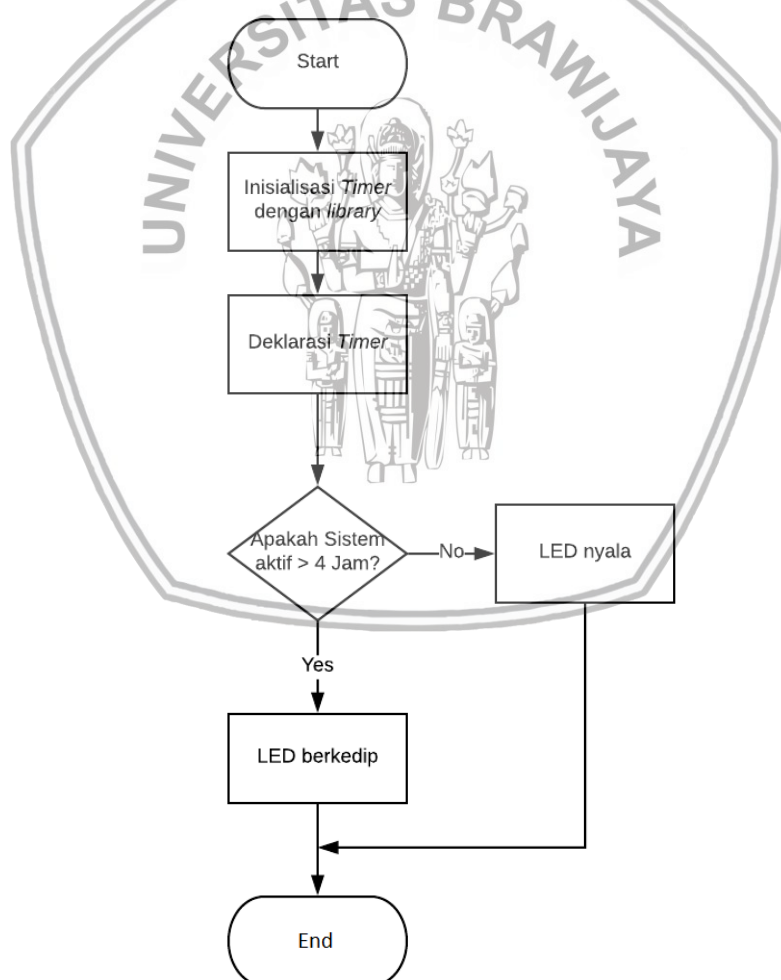


Gambar 5.9 *Flowchart* Perancangan *Software* Sensor Ultrasonik

Pertama pin-pin akan dideklarasikan terlebih dahulu yang nantinya akan digunakan dalam kode program. Selanjutnya nilai akan masuk dan dibaca oleh sensor ultrasonik lalu nilai tersebut dikonversikan ke dalam satuan *centimeter*. Dalam *centimeter* jika jarak yang dibaca melebihi 100 cm maka sistem akan masuk ke mode *sleep*. Untuk menghidupkannya dibutuhkan *interrupt external* oleh sensor LDR. Jika sensor ultrasonik membaca nilai kurang dari 45 cm maka LED akan berkedip sebaliknya LED nyala.

5.1.2.3 Perancangan *Software Timer*

Perancangan *software timer* merupakan rancangan perangkat lunak untuk *timer* pada sistem. Sistem akan menghitung mundur waktu yang mengindikasikan sudah berapa lama sistem tersebut dijalankan. Jika sistem sudah berjalan lebih dari 4 jam maka LED akan berkedip. Diagram alir *timer* dapat dilihat pada gambar 5.10.

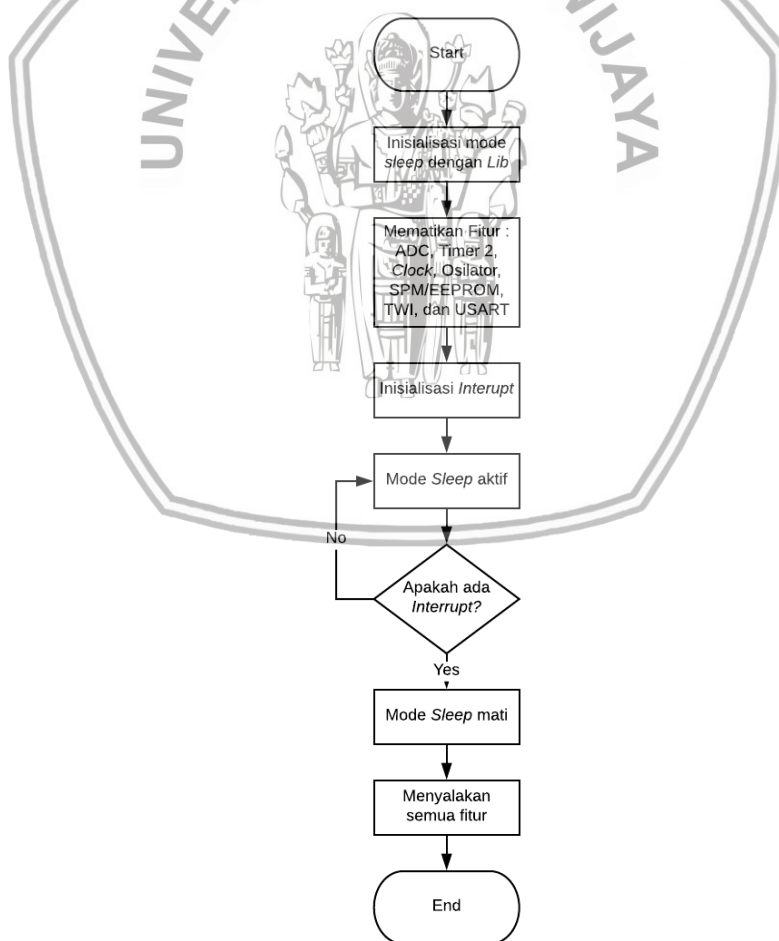


Gambar 5.10 Perancangan *Software Timer*

Pada *timer*, pin-pin akan di deklarasikan terlebih dahulu lalu dengan menggunakan *library* maka kode program *timer* dapat dibuat. Jika timer telah menghitung waktu dalam satuan sekon melebihi 4 jam. Maka LED akan berkedip, sebaliknya LED akan nyala.

5.1.2.4 Perancangan *Software Low Power*

Perancangan *software low power* merupakan tujuan utama dari penelitian ini, yaitu merancang sistem sedemikian rupa sehingga sistem akan menjadi sistem yang hemat penggunaan daya energi dan sumber daya tanpa menghilangkan fungsi utama sistem. Jenis *low power sleep power down* digunakan untuk menekan pengeluaran arus sistem dimana penekanan terjadi ketika sistem sedang tidak digunakan, atau dalam keadaan *sleep*. *Low power sleep power down* akan mematikan beberapa fungsional dari ATmega328P sesuai dengan ketentuan mikontroler tersebut hingga sistem dinyalakan kembali dengan *diinterrupt* oleh sensor LDR. Adapun diagram alir dari sistem perancangan *low power* dapat dilihat pada gambar 5.11.



Gambar 5.11 Perancangan *Software Low Power*

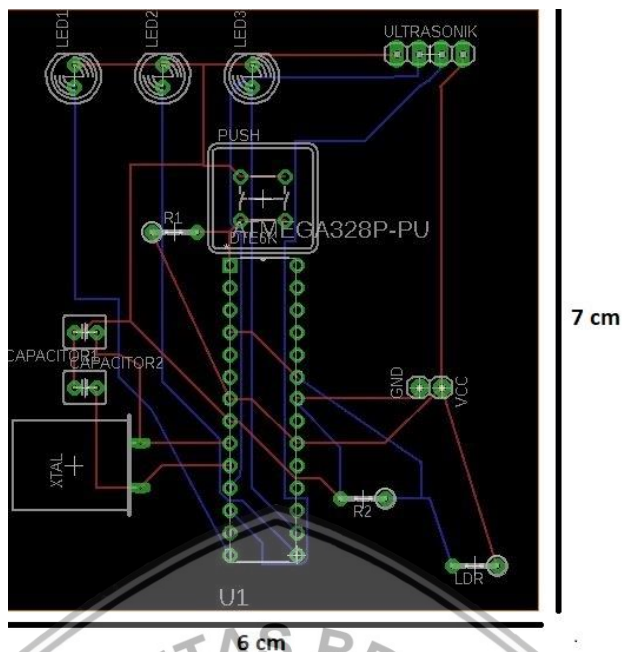
Untuk *software lowpower* pertama, mode *sleep* akan diinisialisasikan dengan *library lowpower*. Selanjutnya dari sistem akan mematikan beberapa fungsional seperti ADC, Timer, Clock, Osilator, SPM/EEPROM, TWI dan USART pada ATmega328P sesuai dengan ketentuan *datasheet*. Dalam keadaan *sleep* kode program juga menginisialisasikan *interrupt* agar untuk digunakan sebagai *interrupt external* yang membangunkan sistem. Jika terjadi *interrupt* maka mode *sleep* akan mati jika tidak terjadi *interrupt* maka sistem akan tetap *sleep*. Jika sudah diaktifkan kembali maka sistem juga ikut mengaktifkan fungsional yang sebelumnya telah dimatikan.

5.1.3 Perancangan *Prototype*

Setelah merancang *hardware* dan *software*, berikutnya adalah perancangan *prototype* atau bentuk purwarupanya. Tujuan dari perancangan *prototype* adalah agar nantinya ketika sistem diimplementasikan rangkaian dapat dibuat dengan baik serta dapat memberikan gambaran seperti apa sistem yang akan dibuat. Adapun *prototype* yang akan dirancang ada 2, yaitu perancangan PCB dan perancangan akirlik

5.1.3.1 Perancangan PCB

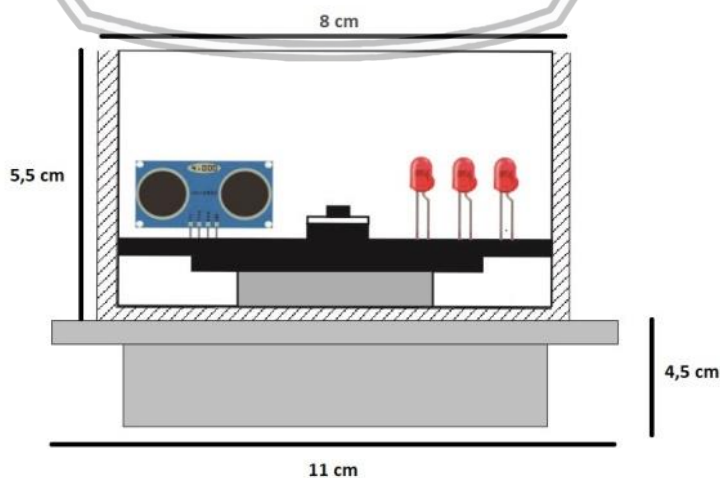
PCB adalah *Printed Circuit Board*, jadi tujuan dari perancangan PCB adalah agar rangkaian sistem dapat tertata dengan baik, sistem masih dapat bekerja dengan baik, dan dapat mencegah adanya *error*. Jalur-jalur pada rangkaian PCB disesuaikan dengan jalur-jalur pada skematik pada gambar 5.2 sebelumnya, sehingga rangkaian PCB akan memiliki kesamaan dengan rangkaian ketika pada *project board*. Rangkaian PCB berukuran 7 cm x 6 cm (P x L), ukuran ini cukup kecil sehingga akan mudah diimplementasikan nantinya. Gambar 5.12 merupakan skematik perancangan PCB.



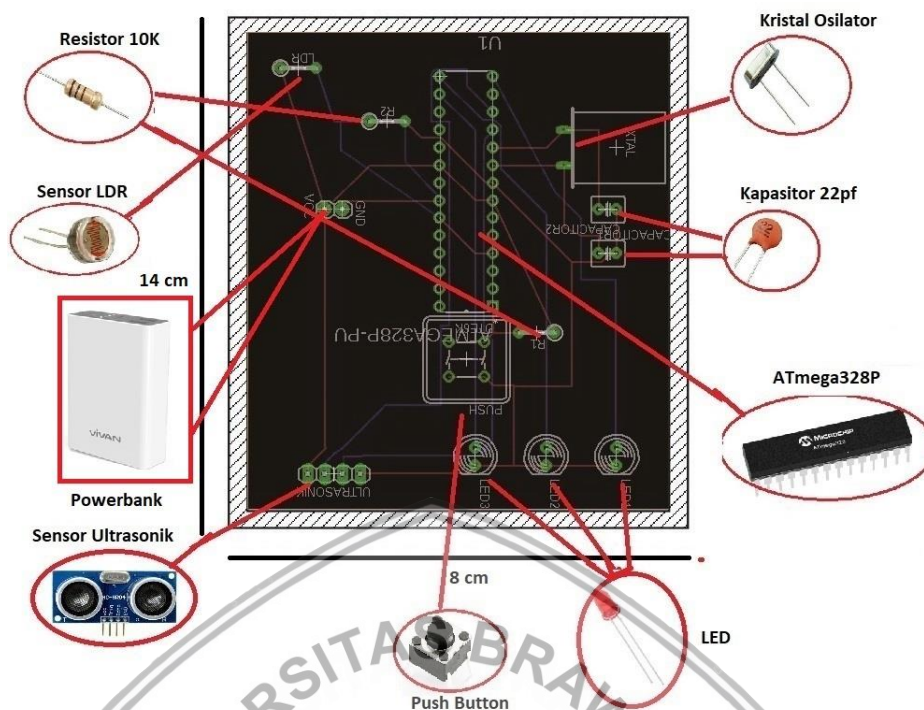
Gambar 5.12 Skematik Perancangan PCB

5.1.3.2 Perancangan Aklirik

Perancangan aklirik merupakan tempat yang nantinya akan berisi PCB buatan ketika diimplementasikan. Aklirik ini akan ditempatkan di atas monitor dengan cara menjepit ke layar monitor dengan penjepit yang telah ditempel dengan aklirik. Oleh karena itu, sistem akan menjadi portable, dimana sistem akan bisa dibawa kemana-mana, ditempatkan pada monitor LCD manapun, bisa ditempatkan di sisi atas, bawah, atau samping. Aklirik berbentuk persegi panjang dengan ukuran 14 cm x 8 cm x 5,5 cm (P x L x T). Gambar 5.13 merupakan gambaran perancangan aklirik.



(a)



Gambar 5. 13 Perancangan Akilirik Tampak Depan (a) Tampak Atas (b)

5.2 Implementasi Sistem

Implementasi Sistem dilakukan berdasarkan dari perancangan *hardware* sistem, perancangan *software* sistem, serta perancangan *prototype* sistem pada sub-bab sebelumnya. Pada sub-bab ini akan dijabarkan mengenai spesifikasi sistem secara *hardware*, *software*, dan *prototype* dari sistem ini sendiri.

5.2.1 Implementasi Hardware

Implementasi *hardware* mengikuti pada sub-bab perancangan sebelumnya, dimana implementasi akan dimulai dari implementasi perangkat catu daya untuk ATmega328P, lalu implementasi sensor LDR pada ATmega328P untuk membaca nilai intensitas cahaya dan juga sebagai *interrupt* ketika sistem *sleep*, implementasi sensor ultrasonik pada ATmega328P untuk membaca nilai jarak dan juga sebagai pemicu *sleep*, dan implementasi LED pada ATmega328P untuk luaran dari sistem. Implementasi sistem ini akan menjadi gambaran sistem ketika sistem siap digunakan, yaitu sistem akan membaca kadar intensitas cahaya lalu menentukan luarannya, sistem akan membaca jarak antara pengguna dan komputer lalu menentukan luarannya, serta sistem dapat *sleep* ketika tidak digunakan lalu *wake* ketika terjadi *interrupt*. Seluruh *hardware* yang telah disebutkan di atas diimplementasikan dalam bentuk PCB yang berukuran 7

cm x 6 cm (P x L) dimana gambar secara keseluruhan PCB, dapat dilihat pada gambar 5.14.

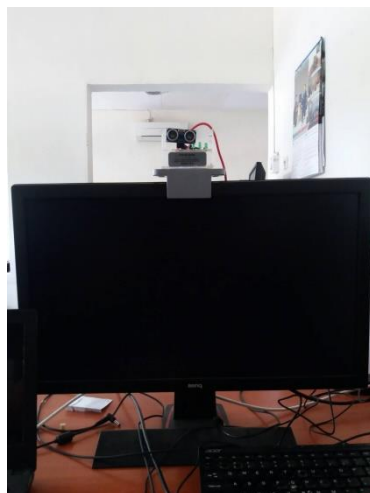


Gambar 5. 14 Implementasi *Hardware* PCB

Setelah diimplementasikan dalam bentuk PCB, berikutnya sistem akan diimplementasikan ke dalam bentuk *prototype* dimana akrilik yg berukuran 14 cm x 8 cm x 5,5 cm (P x L x T) yang sudah ditempel dengan penjepit tersebut akan menjadi wadah dari PCB. Akrilik ini nantinya dapat dijepit pada layar monitor sehingga dapat sistem dapat berjalan sesuai dengan prinsip kerjanya. Bentuk *prototype* dapat dilihat pada gambar 5.15 dan 5.16



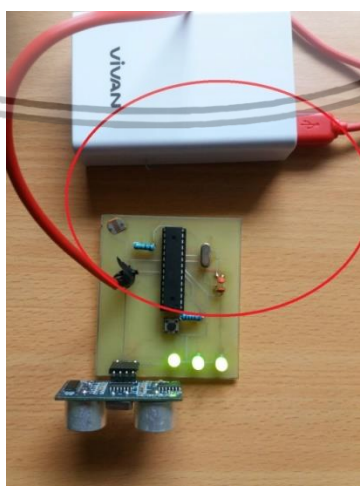
Gambar 5. 15 Implementasi *Hardware Prototype* Tampak Atas



Gambar 5. 16 Implementasi Hardware Prototype Tampak Depan

5.2.1.1 Implementasi *Hardware* Catu Daya

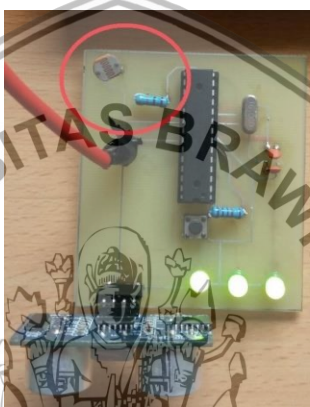
Pada implementasi *hardware* catu daya akan menghasilkan berupa rangkaian-rangkaian yang nantinya akan menghasilkan sumber daya kepada sistem. Sumber daya yang digunakan adalah *powerbank* yang nantinya akan menjadi sumber daya utama untuk menghidupkan sistem. Sistem yang telah dirancang dan ditempatkan pada PCB tetap memiliki pin-head khusus untuk *powerbank* yang nantinya digunakan untuk menguji arus yang dihasilkan sistem pada saat aktif maupun pada saat *sleep*. Kabel positif warna merah dari *powerbank* dihubungkan pada VCC sistem dan kabel negatif warna hitam dari *powerbank* akan dihubungkan pada GND sistem. Gambaran hasil implementasi *hardware* catu daya dapat dilihat pada gambar 5.17 tepatnya yang dilingkari merah.



Gambar 5.17 Implementasi Catu Daya

5.2.1.2 Implementasi *Hardware* Sensor LDR

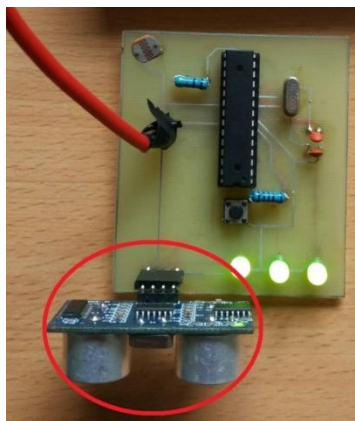
Pada implementasi *hardware* sensor LDR akan menghasilkan berupa rangkaian-rangkaian yang nantinya akan membaca nilai kadar intensitas cahaya dalam satuan lux. Kadar nilai intensitas cahaya ini akan menjadi penentu luaran dari sistem. Jika terlalu gelap maka LED akan berkedip, jika tidak maka LED akan nyala. Sensor LDR juga sebagai *interrupt* untuk mengaktifkan kembali sistem jika dalam keadaan *sleep*. Penempatan sensor LDR diletakan pada bagian pojok kanan bawah PCB yang bertujuan agar sensor LDR dapat menangkap cahaya lebih baik karena jauh dari komponen-komponen lain dengan begitu sensor LDR akan bekerja dengan baik. Hasil implementasi *hardware* sensor LDR dapat dilihat pada lingkaran merah gambar 5.18.



Gambar 5. 18 Implementasi *Hardware* Sensor LDR

5.2.1.3 Implementasi *Hardware* Sensor Ultrasonik

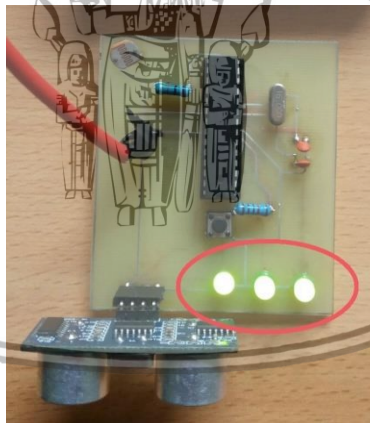
Pada implementasi *hardware* sensor ultrasonik akan menghasilkan berupa rangkaian-rangkaian yang nantinya akan membaca nilai jarak antara pengguna dengan komputer dalam satuan *centimeter*. Kadar nilai jarak ini akan menjadi penentu luaran sistem. Jika pengguna terlalu dekat dengan monitor yang berarti nilai yang dibaca sensor ultrasonik kecil maka lampu LED akan berkedip, sebaliknya lampu LED akan nyala. Sensor ultrasonik juga akan menjadi pemicu sistem untuk *sleep* ketika sensor ultrasonik membaca nilai jarak tertentu. Sensor ultrasonik ditempatkan pada bagian pojok atas kanan PCB yang bertujuan agar sensor ultrasonik dapat memantulkan sinyal *trigger* dan kembali ke *echo* tanpa ada yang menghalangi. Berikut merupakan implementasi *hardware* sensor ultrasonik yang ditunjukkan pada lingkaran merah gambar 5.19.



Gambar 5.19 Implementasi *Hardware* Sensor Ultrasonik

5.2.1.4 Implementasi *Hardware* LED

Pada implementasi *hardware* LED adalah rangkaian-rangkaian yang menjadi luaran dari sistem ini. LED akan menyala atau mati ketika LED sensor-sensor mencapai nilai-nilai tertentu. Selain sensor, *timer* yang ada pada sistem juga membutuhkan LED sebagai luaran ketika sistem telah berjalan melewati batas waktu yang telah ditentukan. Penempatan LED diletakkan di sebelah kiri sensor ultrasonik yang bertujuan agar LED dapat terlihat dengan jelas ketika aktif/berkedip. Gambar 5.20 menunjukkan gambar implementasi *hardware* LED



Gambar 5.20 Implementasi *Hardware* LED

5.2.2 Implementasi *Software*

Implementasi *software* merupakan rangkaian untuk menjalankan proses-proses secara *software* sesuai dengan perancangan *software* pada sub bab sebelumnya. Mengikuti *flowchart-flowchart* yang sudah disusun pada sub bab sebelumnya, maka kode program dapat diimplementasikan kedalam *software* dengan bantuan *software* arduino IDE. Penjelasan mengenai masing-masing jenis kode program dijelaskan pada sub bab ini.

5.2.2.1 Implementasi *Software Library*

Dalam mengimplementasikan kode program, peneliti menggunakan *library* yang akan digunakan ATmega328P untuk mempermudah peneliti membuat kode program sistem. *Library* tersebut berupa *library* untuk *lowpower* yang ditunjukkan pada baris 3 dimana akan digunakan untuk penghematan sumberdaya dengan cara men-*sleep* sistem ketika sistem tidak digunakan lalu menghidupkan/*wake* ketika sistem ingin digunakan kembali. *Library* lain yang digunakan adalah *library timer* yang ditunjukkan pada baris 1 dan 2, berfungsi sebagai *timer* berapa lama sistem telah berjalan. Berikut merupakan implementasi *software library* yang dapat dilihat pada tabel 5.6

Tabel 5.6 Implementasi *Software Library*

<i>Software Library</i>	
1	#include <Event.h>
2	#include <Timer.h>
3	#include <LowPower.h>

5.2.2.2 Implementasi *Software Pin-Pin ATmega328P*

Implementasi *software* pin-pin ATmega328P meliputi inisialisasi pin-pin yang digunakan oleh sistem. Adapun pin-pin ATmega328P yang digunakan adalah, pin digital 8 9 10 untuk LED, pin digital 5 6 untuk sensor ultrasonik, pin analog 0 dan digital 2 untuk sensor LDR. Selain inisialisasi pin-pin, kode program juga menginisialisasi variabel global dimana variabel ini dapat digunakan dimana saja walaupun kode program terdapat di dalam suatu fungsi. Tabel 5.7 menunjukkan gambaran implementasi *software* pin-pin ATmega328P. Untuk proses perhitungan *timer* dilihat pada kode program baris 9.

Tabel 5.7 Implementasi *Software Pin-Pin ATmega328P*

<i>Software Pin ATmega328P</i>	
1	#define led1 8 // LED Cahaya
2	#define led2 9 // LED Jarak
3	#define led3 10 // LED Waktu
4	
5	float readA0;
6	int trigPin = 6; //Trig - green Jumper
7	int echoPin = 5; //Echo - yellow Jumper
8	long duration, cm;
9	int wakeUpPin = 2;
10	int S = 59; // Sekon
11	int M = 59; // Menit
12	int H = 3; // Jam
13	int S2= 5;
14	long period1 = 500 ;
15	long period2 = 500 ;
16	long time;
17	Timer t, t2;

5.2.2.3 Implementasi Software Low Power

Implementasi *software low power* meliputi kode program-kode program yang dapat diproses berdasarkan *library low power*. Pada kode program ini akan dideklarasikan mengenai hal yang akan dilakukan program untuk membuat sistem *sleep*. Kode program ini meliputi mematikan beberapa fungsi ATmega328P saat *sleep* sesuai dengan ketentuan ATmega328P, mematikan LED ketika sedang hidup, lalu memasang *interrupt* jika suatu saat akan terjadi *interrupt* pada pin digital 2 dan melepas *interrupt* jika sudah terjadi *interrupt* pada pin digital 2. Peneliti menggunakan *low power sleep power down* dikarenakan mode *low power* ini memiliki daya hemat yang lebih baik dari pada *low power* lainnya. Terlihat pada baris 10, fungsi dari *sleep power down* hanya mematikan fungsional ADC dan BOD. Namun sebenarnya, *sleep power down* mematikan fungsional lain seperti SPI, USART, TIMER, dan TWI. Berikut merupakan tabel 5.8 implementasi *software low power*.

Tabel 5.8 Implementasi Software Low Power

Software Low Power	
1	void wakeUp(){
2	
3	}
4	
5	void lowpower(){
6	attachInterrupt(0, wakeUp, FALLING);
7	digitalWrite(led1, LOW);
8	digitalWrite(led2, LOW);
9	digitalWrite(led3, LOW);
10	LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF);
11	detachInterrupt(0);
12	
13	}

5.2.2.4 Implementasi Software Sub Program

Implementasi *Software Sub Program* merupakan implementasi kode program *setup* dimana fungsi dari kode program ini untuk mendeklarasikan pin-pin yang sudah diinisialisasikan sebelumnya agar pin tersebut dapat bekerja sesuai dengan prinsipnya. Selain pin, pada kode program ini terdapat kode program untuk mempersiapkan fungsi *timer* yang nantinya akan menghitung waktu. Kode program ini juga hanya akan berjalan sekali. Kode program implementasi *software sub program* dapat dilihat pada tabel 5.9.

Tabel 5.9 Implementasi Software Sub Program

Software Sub Program	
1	void setup() {
2	pinMode(trigPin, OUTPUT);
3	pinMode(echoPin, INPUT);
4	pinMode(wakeUpPin, INPUT);
5	pinMode(led1, OUTPUT);
6	pinMode(led2, OUTPUT);
7	pinMode(led3, OUTPUT);
8	t.oscillate(led1, period1, HIGH);
9	t2.oscillate(led2, period2, HIGH);
10	Serial.begin (9600);
11	}

5.2.2.5 Implementasi Software Main Program

Implementasi *software main program* merupakan implementasi kode program pada *loop* dimana kode program ini berisi kode program sensor LDR dengan kondisi apa yang terjadi ketika lux yang dibaca kurang dari ketentuan, kode program sensor ultrasonik dengan kondisi apa yang terjadi ketika jarak yang dibaca tidak sesuai ketentuan, dan kode program untuk *timer* itu sendiri.

Software sensor LDR akan menghitung dan mengkalibrasi nilai ADC yang masuk ke ATmega328P dari sensor LDR untuk dikonversikan ke satuan Lux. Keluaran dari jumlah lux yang dihitung oleh sensor LDR bervariasi karena sensor LDR sendiri termasuk sensor yang sensitif terhadap pencahayaan. Jumlah lux akan mengecil ketika ruangan dalam keadaan gelap dan jumlah lux akan membesar ketika ruangan dalam keadaan terang. Baris 3 sampai 13 menunjukkan kode program untuk membaca sensor LDR lalu mengeluarkan luaran.

Software sensor ultrasonik akan menghitung jarak antara pengguna dengan sistem lalu dikonversikan ke dalam satuan *centimeter*. Keluaran dari sensor ultrasonik tergantung dari lamanya sinyal yang dikirimkan dari pin *trigger* dapat ditangkap oleh pin *echo* dimana lamanya sinyal tersebut akan menjadi berapa jarak antara pengguna dengan sistem. Perhitungan dan kode program dari sensor ultrasonik berserta luarannya dapat dilihat pada kode program baris 15 sampai 36.

Software timer merupakan kode program yang menentukan batas waktu dari sistem. Sistem pada kode program, sistem akan menghitung mundur dengan menggunakan *decrement* yang ditandai pada baris 39 – 49. Baris selanjutnya adalah LED yang berkedip ketika timer sudah menghitung habis waktu. Secara keseluruhan, kode program implementasi *software main program* dapat dilihat pada tabel 5.10.

Tabel 5.10 Implementasi Software Main Program

Software Main Program	
1	void loop(){
2	//-----Hitungan Intensitas Cahaya-----
3	readA0=analogRead(A0);
4	float Vout0 = readA0*0.0048828125;
5	float lux0 = 10.0 * ((5-Vout0)/Vout0);
6	float lux = 500.0 / lux0;
7	float lux1 = lux ;//- 22.14;
8	if (lux1 <= 100.0){
9	t.update();
10	}
11	else{
12	digitalWrite(led1, HIGH);
13	}
14	//-----Hitungan Jarak-----
15	digitalWrite(trigPin, LOW);
16	delayMicroseconds(5);
17	digitalWrite(trigPin, HIGH);
18	delayMicroseconds(10);
19	digitalWrite(trigPin, LOW);
20	pinMode(echoPin, INPUT);
21	duration = pulseIn(echoPin, HIGH);
22	cm = (duration/2) / 29.1;
23	if (cm >= 100){
24	S2--;
25	delay(1000);
26	if (S2<0 && cm >=100){
27	lowpower();
28	S2=5;
29	}
30	}
31	else if (cm <=45){
32	t2.update();
33	}
34	else{
35	digitalWrite(led2, HIGH);
36	}
37	//-----Hitungan Timer-----
38	digitalWrite(led3, HIGH);
39	S--;
40	delay(1000);
41	
42	if (S<0){
43	M--;
44	S=59;
45	}
46	if (M<0){
47	H--;
48	M=59;
49	}
50	if (H<0){
51	digitalWrite(led3, LOW);
52	delay(250);
53	digitalWrite(led3, HIGH);
54	delay(500);}}

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab 6 pengujian akan membahas mengenai pembahasan dari masing-masing fungsionalitas sistem yang terdiri dari pengujian fungsionalitas sensor ultrasonik, pengujian fungsionalitas sensor LDR, pengujian fungsionalitas *timer*, pengujian arus ketika *sleep*, dan pengujian waktu ketika *wake*.

6.1 Pengujian Fungsionalitas Sensor Ultrasonik

Pengujian fungsionalitas sensor ultrasonik merupakan pengujian yang dilakukan dengan menggunakan sensor ultrasonik dimana akan dibaca jarak antara pengguna dengan sistem. Pengujian dengan membandingkan hasil bacaan sensor ultrasonik pada *serial monitor* dengan hasil bacaan secara manual yaitu dengan meteran. Hasil dari pengujian akan dibandingkan lalu dianalisis.

6.1.1 Tujuan

Untuk menguji sensor ultrasonik yang menghasilkan jarak dari pengguna ke sistem dalam satuan *centimeter* apakah sesuai dengan alat ukur manual yaitu meteran atau tidak. Dari sini akan diketahui tingkat akurasi sensor ultrasonik dalam membaca nilai.

6.1.2 Prosedur Pengujian

Untuk dapat menguji akurasi sensor ultrasonik dibutuhkan prosedur-prosedur pengujian yang dapat dilihat sebagai berikut:

1. Menyiapkan alat-alat yang dibutuhkan, yaitu mikrokontroler ATmega328, sensor ultrasonik, dan kabel jumper.
2. Sambungkan pin Trig dan Echo pada sensor ultrasonik pada pin digital 6 dan pin digital 5 ATmega328P serta VCC dan GND pada sensor ultrasonik pada pin VCC dan pin GND ATmega328P.
3. Buka *software* arduino IDE lalu ketik *source code* untuk membaca jarak lalu *upload* ke dalam mikrokontroler ATmega328P.
4. Letakkan sensor ultrasonik di atas monitor lalu nyalakan *serial monitor* pada *software* arduino IDE.
5. Amati dan catat setiap keluaran yang dihasilkan sensor ultrasonik pada *serial monitor*.
6. Lakukan pada berbagai kondisi, seperti misalnya pengguna yang mendekat, atau pengguna yang menjauh dari sistem dan ulangi dari sampai 10 kali.
7. Bandingkan hasil bacaan sensor ultrasonik dengan meteran

8. Menentukan presentase *error* dan akurasi sensor dengan menggunakan **persamaan 6.1** di bawah :

$$PE = \frac{SB}{PB} \times 100\% \quad (6.1)$$

Berdasarkan **persamaan 6.1** terdapat keterangan yang dapat diketahui sebagai berikut :

PE = Presentase *error*

SB = Selisih nilai pembacaan

PB = Pembacaan Meteran

Untuk menghitung selisih nilai pembacaan dapat dilihat pada **persamaan 6.2** di bawah :

$$SB = PB - PS \quad (6.2)$$

Berdasarkan **persamaan 6.2** terdapat keterangan yang dapat diketahui sebagai berikut :

PS = Pembacaan sensor

Untuk menghitung presentase akurasi sensor ultrasonik dapat dilihat pada **persamaan 6.3** di bawah :

$$\text{Presentase akurasi} = 100\% - \text{Presentase error} \quad (6.3)$$

6.1.3 Hasil dan Analisis

Pengujian sensor ultrasonik dilakukan pada 1 orang pengguna saja namun diuji selama 10 kali. Berikut merupakan tampilan nilai yang dibaca oleh sensor ultrasonik melalui *serial monitor* yang ditunjukkan pada gambar 6.1.



Gambar 6.1 Pengujian Sensor Ultrasonik pada *Serial Monitor*

Pada gambar 6.1, nilai dibaca oleh sensor ultrasonik dalam jeda waktu 1 detik lalu hasil bacaan tersebut ditampilkan pada *serial monitor* yang bertujuan untuk mempermudah proses pengujian. Dapat dilihat pada gambar 6.1, hasil bacaan sensor ultrasonik memiliki *range* dari 39 – 50 cm. Setelah diukur melalui *serial monitor* berikutnya diukur dengan alat ukur meteran yang ditunjukkan pada gambar 6.2.



Gambar 6.2 Pengujian Sensor Ultrasonik pada Meteran

Pada gambar 6.2, hasil yang didapatkan dari mengukur jarak antara pengguna dengan sistem dengan alat ukur meteran adalah 40 cm. Hasil tersebut dapat berubah karena sistem masih diuji dengan meteran sebanyak 9 kali lagi. Setelah didapat hasil pengujian baik dari *serial monitor* maupun dari alat ukur meteran, maka dapat dihitung presentase *error*-nya yang dapat dilihat pada tabel 6.1.

Tabel 6.1 Hasil Pengujian Sensor Ultrasonik

Pengujian Ke-	Panjang dari Meteran (CM)	Panjang dari <i>Serial Monitor</i> (CM)	Selisih Panjang	Presentase <i>Error</i> (%)
1	46	45	1	2,17
2	49	45	4	8,16
3	50	50	0	0
4	50	50	0	0
5	49	48	1	2,04
6	46	45	1	2,17
7	44	44	0	0
8	44	42	2	4,54
9	44	44	0	0
10	43	43	0	0
Rata-Rata	46,5	45,6	0,9	1,93

Pada tabel 6.1 dapat dilihat bahwa sensor ultrasonik memiliki presentase *error* sebesar 1,93 % jika dibandingkan dengan alat ukur meteran. Hasil presentase tersebut di dapat dari **persamaan 6.1** di atas, sedangkan sensor ultrasonik memiliki akurasi sebesar 98,07 % dimana hasil akurasi tersebut dapat dilihat pada **persamaan 6.3** di atas. Hal ini menunjukkan bahwa dalam pembacaan nilai oleh sensor ultrasonik, terdapat banyak faktor yang mempengaruhi seperti misalnya adalah waktu jeda sensor, jarak antara pengguna dan sistem, ataupun pengguna yang bergerak sedikit.

6.2 Pengujian Fungsionalitas Sensor LDR

Pengujian fungsionalitas sensor LDR merupakan pengujian yang dilakukan dengan menggunakan sensor LDR dimana akan dibaca kadar intensitas cahaya di dalam ruangan. Pengujian dengan membandingkan hasil bacaan sensor LDR pada *serial monitor* dengan hasil bacaan secara manual yaitu dengan Luxmeter. Hasil dari pengujian akan dibandingkan lalu dianalisis.

6.2.1 Tujuan

Untuk menguji sensor LDR yang menghasilkan kadar intensitas cahaya dalam suatu ruangan dalam satuan lux, apakah sesuai dengan alat ukur manual yaitu luxmeter atau tidak. Dari sini akan diketahui tingkat akurasi sensor LDR dalam membaca nilai.

6.2.2 Prosedur Pengujian

Untuk dapat menguji akurasi sensor LDR dibutuhkan prosedur-prosedur pengujian yang dapat dilihat sebagai berikut:

1. Menyiapkan alat-alat yang dibutuhkan, yaitu mikrokontroler ATmega328P, sensor LDR, resistor 10k ohm, dan kabel *jumper*.
2. Menyusun rangkaian pembagi tegangan pada sensor LDR, dimana kutub (+) sensor LDR disambungkan pada VCC mikrokontroler ATmega328P, kutub (-) sensor LDR disambungkan oleh pin A0 pada mikrokontroler ATmega328P dan resistor 10k ohm, resistor 10k ohm berikutnya disambungkan pada pin GND mikrokontroler ATmega328P.
3. Buka *software* arduino IDE lalu ketik *source code* untuk membaca kadar intensitas cahaya lalu *upload* ke dalam mikrokontroler ATmega328P.
4. Letakkan sensor LDR di atas monitor lalu nyalakan *serial monitor* pada *software* arduino IDE.
5. Amati dan catat setiap keluaran yang ditampilkan pada *serial monitor*.
6. Amati juga keluaran dari alat ukur Luxmeter.
7. Bandingkan hasil keluaran Luxmeter dengan luaran *serial monitor*.
8. Menentukan presentase *error* dan akurasi dengan menggunakan **persamaan 6.1, persamaan 6.2, dan persamaan 6.3.**

6.2.3 Hasil dan Analisis

Pengujian sensor LDR dilakukan pada ruangan tertutup namun cukup pencahayaan, yaitu diuji pada siang hari. Sensor LDR diuji sebanyak 10x dengan 1 pengguna. Berikut merupakan tampilan nilai yang dibaca sensor LDR pada *serial monitor* ditunjukkan pada gambar 6.3.



Gambar 6.3 Pengujian Sensor LDR pada *Serial Monitor*

Pengujian sensor LDR dilakukan pada jeda waktu 1 detik lalu hasil nilai bacaan ditampilkan pada *serial monitor* untuk mempermudah proses pengujian. Dapat dilihat pada gambar 6.3, hasil bacaan sensor LDR memiliki *range* sebesar 272 – 278 lux. Setelah diukur dan ditampilkan pada *serial monitor* berikutnya diukur dengan alat ukur Luxmeter yang ditunjukkan pada gambar 6.4.



Gambar 6.4 Pengujian Sensor LDR pada Luxmeter

Pada gambar 6.4, hasil yang didapatkan dari mengukur kadar intensitas cahaya di dalam ruangan adalah sebesar lux. Hasil tersebut dapat berubah karena sistem masih diuji dengan Luxmeter sebanyak 9 kali lagi. Setelah didapat hasil pengujian baik dari *serial monitor* maupun dari alat ukur Luxmeter, maka dapat dihitung presentase *error*-nya yang dapat dilihat pada tabel 6.2.

Tabel 6.2 Hasil Pengujian Sensor LDR

Pengujian Ke-	Intensitas Cahaya dari Luxmeter (Lux)	Intensitas Cahaya dari <i>Serial Monitor</i> (Lux)	Selisih Intensitas	Presentase <i>Error</i> (%)
1	308	295	13	4,22
2	306	295	11	3,59
3	309	298	11	3,55
4	304	295	9	2,96
5	307	298	9	2,93
6	307	298	9	2,93
7	308	295	13	4,22
8	307	295	12	3,90
9	309	296	13	4,20
10	305	295	10	3,27
Rata-rata	307	296	11	3,58

Pada tabel 6.2 dapat dilihat bahwa sensor LDR memiliki presentase *error* sebesar 3,58 % jika dibandingkan dengan alat ukur Luxmeter. Hasil presentase tersebut di dapat dari **persamaan 6.1** di atas, sedangkan sensor LDR memiliki akurasi sebesar 96,42 % dimana hasil akurasi tersebut dapat dilihat pada **persamaan 6.3** di atas. Hal ini menunjukkan bahwa dalam pembacaan nilai oleh sensor LDR, terdapat banyak faktor yang mempengaruhi seperti misalnya adalah keadaan ruang disekitar, tingkat sensitifitas Luxmeter dan sensor LDR, dan tingkat cahaya yang masuk dari luar ruangan.

6.3 Pengujian Fungsionalitas *Timer*

Pengujian fungsionalitas *timer* merupakan pengujian yang dilakukan dengan menggunakan *timer* yang ditetapkan dalam *source code* sistem. *Timer* ditetapkan selama 4 jam pada *source code* dimana pengujian dengan membandingkan waktu sebenarnya dengan waktu pada sistem. Hasil dari pengujian akan dibandingkan dan lalu dianalisis.

6.3.1 Tujuan

Untuk menguji *timer* selama 4 jam, apakah nantinya akan sesuai dengan waktu sebenarnya atau tidak. Dari sini dapat diketahui keberhasilan fungsional *timer* sistem. Keberhasilan fungsional *timer* ditandai dengan berkedipnya lampu LED.

6.3.2 Prosedur Pengujian

Untuk dapat menguji fungsionalitas *timer* dibutuhkan prosedur-prosedur pengujian yang dapat dilihat sebagai berikut:

1. Menyiapkan alat-alat yang dibutuhkan, yaitu mikrokontroler ATmega328P dan lampu LED.
2. Menyambungkan kutub positif lampu LED pada pin digital 10 pada ATmega328P.
3. Buka *software* Arduino IDE lalu ketik dan *upload source code timer* ke ATmega328P.
4. Letakkan sistem di atas monitor lalu biarkan sistem menyala selama 4 jam.
5. Catat waktu yang diperlukan sistem untuk membuat lampu LED berkedip

6.3.3 Hasil dan Analisis

Pengujian waktu *timer* sistem dapat dilakukan dimana saja, asalkan sistem dapat hidup selama 4 jam untuk menguji keberhasilan *timer* sistem. Selama pengujian waktu *timer*, peneliti hanya membiarkan sistem hidup tanpa melakukan aktifitas lain pada sistem, seperti lampu LED ultrasonik yang berkedip, lampu LED sensor LDR yang berkedip, ataupun sistem *sleep*. Berikut merupakan tabel keberhasilan fungsional *timer* sistem yang ditunjukkan pada tabel 6.3

Tabel 6.3 Keberhasilan Fungsionalitas *Timer* Sistem

Pengujian Ke-	<i>Timer</i> Sebenarnya (JJ:MM:SS)	<i>Timer</i> Sistem (JJ:MM:SS)	Lampu LED Berkedip
1	04:00:00	04:01:10	Ya
2	04:00:00	04:00:38	Ya
3	04:00:00	04:01:20	Ya

Pada tabel 6.3 dapat dilihat bahwa lampu LED berkedip setelah sistem hidup selama lebih dari 4 jam. Sistem tidak tepat untuk mengkedipkan LED saat tepat 4 jam dikarenakan adanya pengaruh dari *source code* program lain yang telah diupload. *Source code* tersebut bisa berupa delay ataupun *timer* lain. Meskipun

begitu, sistem masih dapat menjalankan fungsionalitas *timernya* dengan cukup baik.

6.4 Pengujian Arus ketika *Lowpower Sleep* dan *Wake*

Pengujian arus ketika *lowpower sleep* dan *wake* adalah bagaimana peneliti menguji arus yang digunakan sistem ketika sistem dalam keadaan *sleep* dan arus yang mengalir ketika sistem *wake*. *Sleep* berfungsi untuk mengurangi konsumsi daya yang digunakan sistem ketika sistem sedang tidak digunakan, sehingga penggunaan sumber daya pun akan lebih hemat. Lalu ketika *wake* sistem berjalan normal dimana semua fungsionalitasnya aktif. Adapun pengujian arus ketika *sleep* dan *wake* dapat dilihat di bawah ini.

6.4.1 Tujuan

Tujuan dari pengujian Arus ketika *lowpower sleep* dan *wake* adalah untuk mengetahui besar arus yang digunakan dalam oleh sistem ketika dalam keadaan *sleep* dan besar arus ketika sistem dalam keadaan *wake*. Satuan yang digunakan untuk arus yang digunakan adalah mili ampere atau disingkat sebagai mA.

6.4.2 Prosedur Pengujian

Untuk dapat menguji arus saat *sleep* dibutuhkan prosedur-prosedur pengujian yang dapat dilihat sebagai berikut:

1. Menyiapkan alat-alat yang dibutuhkan yaitu mikrokontroler ATmega328P multimeter, sensor ultrasonik, sensor LDR, dan kabel *jumper*.
2. Sambungkan kaki positif multimeter pada kaki GND ATmega328P lalu kaki negatif multimeter pada kaki GND kabel yang terkoneksi ke *powerbank*.
3. Sambungkan kaki-kaki sensor ultrasonik dan sensor LDR sesuai dengan yang ada pada sub bab pengujian sensor ultrasonik dan sensor LDR
4. Buka *software* Arduino IDE lalu ketik dan *upload source code lowpower* ke ATmega328P.
5. Buat sistem masuk ke keadaan *sleep* lalu amati perubahan arus yang tertera pada multimeter.
6. Catat setiap perubahan arus ketika sistem dalam keadaan *sleep*.
7. *Wake* sistem dengan cara memberikan *interrupt external* .
8. Ulangi langkah 2 lalu amati dan catat setiap perubahan arus ketika sistem dalam keadaan *wake* .

6.4.3 Hasil dan Analisis

Pengujian mekanisme *lowpower* sistem dapat dilakukan ketika sistem dalam keadaan *sleep*. Arus yang tertera pada multimeter ketika sistem *sleep* adalah

arus yang mengalir di dalam sistem. Gambar 6.5 menunjukkan jumlah arus yang mengalir di dalam sistem.



Gambar 6.5 Hasil Pembacaan Arus oleh Multimeter ketika *Sleep*

Selain *sleep*, sistem juga diuji ketika sistem dalam keadaan *wake*. Adapun jumlah arus yang tertera pada multimeter ketika sistem *wake* ditunjukkan pada gambar 6.6.



Gambar 6.6 Hasil Pembacaan Arus oleh Multimeter ketika *Wake*

Pada gambar 6.5 dan 6.6, sistem diujikan dengan multimeter baik dalam keadaan *sleep* maupun *wake*, tidak diujikan sekali. Sistem diujikan sebanyak 5 kali untuk mengetahui perbedaan arus sistem. Tabel 6.4 menunjukkan perbedaan arus yang mengalir ketika sistem *sleep* dan sistem *wake*.

Tabel 6.4 Perbedaan Arus ketika *Sleep* dan *Wake*

Pengujian Ke-	Arus ketika <i>Sleep</i> (mA)	Arus ketika <i>Wake</i> (mA)	Selisih Arus (mA)
1	3,8	170,8	167,0
2	3,9	170,1	166,2
3	3,9	169,9	166,0
4	3,8	170,0	166,2
5	3,9	169,0	165,1
Rata-Rata	3,86	169,96	166,1

Pada tabel 6.4 dapat dilihat perbedaan antara sistem dalam keadaan *sleep* dengan sistem dalam keadaan *wake*. ATmega328P dapat memasuki mode *sleep* ketika sensor ultrasonik tidak mendeteksi adanya objek/pengguna dalam jarak 100 cm. Saat keadaan *sleep*, sistem masih mengalirkan arus dengan rata-rata sebesar 3,86 mA. Sedangkan pada keadaan *wake* sistem mengalirkan arus dengan rata-rata sebesar 169,96 mA. Maka didapatkanlah selisih antara arus ketika sistem dalam keadaan *sleep* dan *wake* yaitu sebesar rata-rata 166,1 mA. Ini menunjukkan bahwa sistem berhasil menurunkan tingkat konsumsi daya dengan memanfaatkan fitur *sleep power down* yang mematikan beberapa fungsionalitas ATmega328P itu sendiri.

6.5 Pengujian Waktu ketika *Wake*

Pengujian waktu ketika *wake* merupakan menguji waktu yang dibutuhkan sistem dari keadaan *sleep* ke keadaan *wake*. Dikarenakan waktu *wake* pada sistem dalam satuan milisekon, maka digunakanlah *serial monitor* beserta modifikasi program untuk mengujinya. Adapun pengujian waktu ketika *wake* ada di bawah ini.

6.5.1 Tujuan

Untuk mengetahui waktu *wake* sistem dalam satuan milisekon. Hasil dari waktu bangun tersebut akan ditampilkan dengan menggunakan *serial monitor*. Dari sini dapat diketahui, seberapa cepat waktu untuk *wake* pada sistem.

6.5.2 Prosedur Pengujian

Untuk dapat menguji waktu yang dibutuhkan sistem untuk *wake* dibutuhkan prosedur-prosedur pengujian yang dapat dilihat sebagai berikut:

1. Menyiapkan ATmega328P, sensor ultrasonik, sensor LDR, dan kabel *jumper*,

2. Sambungkan kaki-kaki sensor ultrasonik dan sensor LDR sesuai dengan yang ada pada sub bab pengujian sensor ultrasonik dan sensor LDR.
3. Buka *software* Arduino IDE lalu modifikasi sedikit pada program utama lalu *upload* ke ATmega328P.
4. Buat sistem menjadi *sleep* lalu *wake* sistem.
5. Amati dan catat berapa waktu yang dibutuhkan agar sistem *wake*.

6.5.3 Hasil dan Analisis

Pengujian waktu ketika *wake* dapat dilakukan ketika sistem dalam keadaan *sleep* lalu sistem tersebut diberikan *interrupt extenal* untuk membangunkan sistem. Dengan begitu jeda antara waktu *sleep* dan *wake* dapat terlihat pada *serial monitor*. Gambar 6.7 menunjukkan hasil waktu *wake* sistem pada *serial monitor*



Gambar 6.7 Hasil Waktu *Wake* pada *Serial Monitor*

Pada gambar 6.7, terdapat banyak waktu dalam milisekon yang terbaca. Namun waktu *wake* pada sistem hanya terjadi pada saat milisekon menunjukkan angka 21. Ini berarti sistem dapat bangun dari keadaan *sleep* dengan hanya membutuhkan waktu sebesar 21 milisekon.

Pada prosedur pengujian nomor 3 terdapat modifikasi program yang khusus untuk melihat waktu yang dibutuhkan untuk sistem ketika *wake*. Untuk modifikasi program, gambar 6.8 dan gambar 6.9 menunjukkan penggalan *source code* untuk menghitung waktu *wake* pada sistem.

```

59 void loop()
60 {
61   double wb= 0;
62   double wm =0;
63   wb = millis();
64   //Hitungan Intensitas Cahaya
65   readA0=analogRead(A0);
66   float Vout0 = readA0*0.0048828125;
67   float lux0 = 10.0 * ((5-Vout0)/Vout0);
68   float lux = 500.0 / lux0;
69   float lux1 = lux ;// - 22.14;

```

Gambar 6.8 Penggalan Source Code Pertama

```

114   if(H<0){
115     digitalWrite(led3, LOW);
116     delay(250);
117     digitalWrite(led3, HIGH);
118     delay(500);
119   }
120   wm = millis();
121   Serial.print("waktu wake : ");
122   Serial.print(wm - wb);
123   Serial.println(" milisekon");
124
125 }

```

Gambar 6.9 Penggalan Source Code Kedua

Pada gambar 6.8, *source code* yang berfungsi untuk menampilkan waktu dalam milisekon ada pada baris ke 61-63, lalu pada gambar 6.9 *source code* yang berfungsi untuk menampilkan waktu dalam milisekon ada pada baris 120-123. Secara sederhana, *source code* di atas hanya menghitung waktu milisekon dengan cara mengurangi waktu milisekon akhir (wm) dengan waktu milisekon awal (wb), sehingga didapatkanlah waktu *wake* pada sistem. Setelah melakukan pengujian selama 5 kali berikutnya adalah mencatat waktu *wake* pada sistem dimana ditunjukkan pada tabel 6.5.

Tabel 6.5 Hasil Pengujian Waktu Wake

Pengujian Ke-	Waktu Wake (ms)
1	21
2	22
3	21
4	24
5	23
Rata-Rata	22,2

Pada tabel 6.5, terdapat 5 pengujian lainnya untuk pengujian waktu *wake* pada sistem. Setelah dilakukan 5 kali pengujian, di dapatlah hasil rata-rata sistem *wake* dalam keadaan *sleep* sebesar 22,2 milisekon. Oleh karena itu ketika tidak ditampilkan dengan *serial monitor*, sistem terlihat dapat *wake* dalam seketika, namun sebenarnya sistem membutuhkan jeda waktu sebesar 22,2 milisekon agar sistem dapat *wake* dari keadaan *sleep* lalu menjalankan fungsionalitasnya sesuai dengan prinsip kerjanya.



DAFTAR PUSTAKA

- Anggraini, Y., 2016. *Dampak Penggunaan Game Online Terhadap Perilaku Remaja*. Jurnal Universitas Sumatera Utara.
- Arduino, 2017. *Arduino*. [online] Arduino Corp. Tersedia di: <<https://www.arduino.cc/en/Main/Software>> [Diakses 20 Desember 2017]
- Azril, 2012. *Luxmeter*. [image online] Tersedia di: <<http://azrilproduction.blogspot.co.id/2012/12/luxmeter.html>> [Diakses 23 Maret 2018]
- depokinstruments, 2006. *HC-SR04. (Ultrasonic Sensor)*. [image online] Tersedia di: <<https://depokinstruments.com/2016/02/23/hc-sr04-ultrasonic-sensor/>> [Diakses 18 Desember 2017]
- Efendi, N. A., 2014. *FAKTOR PENYEBAB BERMAIN GAME ONLINE DAN DAMPAK NEGATIFNYA BAGI PELAJAR*. Naskah Publikasi Universitas Muhammadiyah Surakarta.
- Fauziah, E. R., 2013. *PENGARUH GAME ONLINE TERHADAP PERUBAHAN PERILAKU*. eJournal Ilmu Komunikasi.
- Firmansyah, E. H., 2018. *Implementasi Low Power Mode Pada Perangkat Sistem Pendeteksi Dini Kebocoran Gas Menggunakan ATmega328p*. S1. Universitas Brawijaya.
- Kemendikbud, 2016. *Kemendikbud: 15 Game ini Berbahaya bagi Anak*. [online] Tersedia di: <<https://news.detik.com/berita/d-3198716/kemendikbud-15-game-ini-berbahaya-bagi-anak>> [Diakses 17 Desember 2017]
- Keputusan Menteri Kesehatan Republik Indonesia, 2002. *Tentang Persyaratan Kesehatan Lingkungan Kerja Perkantoran dan Industri Nomor 1405/MENKES/SK/XI/2002*. [pdf] Keputusan Menteri Kesehatan Republik Indonesia. Tersedia di: <https://www.gbcindonesia.org/download/doc_download/41-kepmenkes-no-1405-tahun-2002> [Diakses 17 Desember 2017]

- Kho, D., 2018. *Pengertian LED (Light Emitting Diode) dan Cara Kerjanya*. [image online] Tersedia di: <<https://teknikelektronika.com/pengertian-led-light-emitting-diode-cara-kerja/>> [Diakses 22 Maret 2018]
- Kiswari, A., 2016. *MENGENAL JENIS ALAT UKUR DAN FUNGSINYA*. [image online] Tersedia di: <<https://anastasiakiswari.wordpress.com/2016/05/11/mengenal-jenis-alat-ukur-dan-fungsinya/>> [Diakses 23 Maret 2018]
- Kiswoyo, B., 2017. *pengertian dan fungsi kristal pada rangkaian elektronika*. [image online] Tersedia di: <<https://www.jalankatak.com/id/kristal/>> [Diakses 23 Maret 2018]
- Microchip Technology Inc., 2016. *ATmega328/P - Complete Datasheet*. [online] Tersedia di: <<http://www.microchip.com/wwwproducts/en/ATmega328p>> [Diakses 20 Desember 2017].
- Newzoo, 2017. *Indonesia Gamer 2017*. [online] Newzoo Corp Website. Tersedia di: <<https://newzoo.com/insights/infographics/the-indonesian-gamer-2017/>> [Diakses 12 Desember 2017]
- NPD, 2015. *Mobile Gaming Consumer Trend 2014*. [online] NPD Group Website. Tersedia di: <<https://www.npd.com/wps/portal/npd/us/news/press-releases/2015/average-time-spent-playing-games-on-mobile-devices-has-increased-57-percent-since-2012/>> [Diakses 12 Desember 2017]
- OptoSupply Inc., 2000. *5mm LED*. [online] Tersedia di: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/418174/OPTOSUPPLY/OSOOJ25111A.html>> [Diakses 22 Maret 2018]
- OSHA, 1997. *Working Safely with Video Display Terminals*. [online] U.S. Department of Labor Occupational Safety and Health Administration. Tersedia di: <<http://www.osha.gov/Publications/osha3092.pdf>> [03 Maret 2018]
- Ploeg, H. P., 2012. *Sitting Time and All-Cause Mortality Risk in 222 497 Australian Adults*. [online] Original Investigation. Tersedia di: <<https://jamanetwork.com/journals/jamainternalmedicine/fullarticle/1108810>> [Diakses 20 Desember 2017]

- Purwanto, E., 2016. *Membaca Nilai Kapasitor*. [image online] Tersedia di : <<http://www.edipurwanto2.com/2016/04/membaca-nilai-kondensator-atau-kapasitor.html>> [Diakses pada 23 Maret 2018]
- Santoso, I., 2009. *interaksi manusia dan komputer edisi 2*. [e-book] Yogyakarta: Penerbit Andi. Tersedia di: Google Books <https://books.google.co.id/books?id=_pXa7CvwTC0C&printsec=frontcover&hl=id#v=onepage&q&f=false> [Diakses 12 Desember 2017]
- Sparkfun, 2013. *Voltage Divider*. [image online] Tersedia di: <<https://learn.sparkfun.com/tutorials/voltage-dividers>> [Diakses 22 Maret 2018]
- Turesna, G., Zulkarnain, & Hermawan., 2015. *Pengendali Intensitas Lampu Ruang Berbasis Arduino UNO Menggunakan Metode Fuzzy Logic*. Jurnal Teknik Elektro Universitas Langlang Buana.
- Wind, D., 2016. *Bahaya dan Efek Negatif Kecanduan Game Online bagi Kesehatan*. [online] Kompasiana Corp Website. Tersedia di: <https://www.kompasiana.com/danurwind/bahaya-dan-efek-negatif-kecanduan-game-online-bagi-kesehatan_57d1256f107f610c45d1712b> [Diakses 17 Desember 2017]
- zonaelektro, 2014. *Sensor Cahaya*. [image online] Tersedia di: <<http://zonaelektro.net/sensor-cahaya/>> [Diakses pada 18 Desember 2017]